Rheology of Particle Suspensions

Fresh Concrete, Mortar and Cement Paste with Various Types of Lignosulfonates

Jon Elvar Wallevik[1]
Department of Structural Engineering
The Norwegian University of Science and Technology (NTNU)
NO-7491 Trondheim, Norway

February 2003

[1]jon.wallevik@bygg.ntnu.no

URN:NBN:no-3374

URN:NBN:no-3374

Abstract

Background and Objectives

Ever since civilizations first started to build, the human race has sought a material that can bind stones into a solid mass. After the rediscovery of cement in 1824, the concrete (essentially a mixture of Portland cement, water and aggregates) has become the most commonly used structural material in modern civilizations. The quality of the concrete structure is of course dependent on the quality of each constituent that is used in the concrete mix. However, this is not the only controlling factor. The quality is also much dependent on the rheological properties of the fresh concrete during placement into the formwork or mold. Such rheological properties are best measured with viscometers of various types.

In this work, coaxial cylinders viscometers are used in measuring the rheological properties of concrete, mortar and cement paste (named ConTec **BML** Viscometer 3 and ConTec Viscometer 4). These cement based materials are mixed with various types of plasticizers and superplasticizers (i.e. lignosulfonates). The mechanism of (super)plasticizers is to disperse the cement particles and thus prevent or reduce cement particle coagulation. In this manner they are capable of imparting considerable physical and economical benefits to the cement based materials in terms of workability and workability retention, using moderate amounts of cement even for low w/c mixes.

Experimental Program

Because rheological measurements on concrete requires large resources in terms of test material, labor and time, the complete laboratory test program of this thesis is not carried out with concrete. Rather, mortar and cement paste are also included. This is done in the anticipation that they will simulate the rheological behavior of concrete. By this approach, it is possible to create a larger and more complicated test program. [In Section 6.3.2, the relationship between the mortar and concrete results are discussed].

Three major themes are considered in this thesis, and consequently it is divided into three parts. The **first part** concerns how the different lignosulfonate types changes the rheological properties of the cement based material as a function of temperature and time. This investigation is made with help from the **second part** of this thesis, which identifies some of the parameters p_1, p_2,... affecting the shear viscosity $\eta = \eta(p_1, p_2, \dots)$ of the cement based material. This is done by investigating the thixotropic behavior of cement paste. An experimental error is present during a viscometric measurement on concrete. The error is generated by particle migration. Investigating and compensating for this error constitutes the **third part** of this thesis. For further details about the objectives of each theme, see Section 1.2. The major results of each part will be given shortly.

Being made of granular particles, it is natural to consider concrete, mortar and cement paste as different types of suspensions. However, the normal procedure of generating a shear viscosity function η dependent on the phase volume Φ is not attempted in this thesis. Rather, the approach is to investigate how the different lignosulfonate types changes the rheological behavior as a function of temperature and time, where the phase volume Φ is the same for the compared batches.

Part One: Effects of Lignosulfonates (Time and Temperature Properties)

Both the fresh concrete and mortar are modeled here as a Bingham fluid, characterized by the yield value τ_o and the plastic viscosity μ from a single rheological experiment. When using lignosulfonates, it is demonstrated that the yield value τ_o is much more sensitive to temperature and time, compared to the plastic viscosity μ. Also, the effect of the different lignosulfonate types is more reflected by the measured yield value τ_o, than by the measured plastic viscosity μ; i.e. comparing the effect of two different polymer types gives usually quite different yield values τ_o, while the plastic viscosity μ is roughly the same for the two cases.

In terms of workability and workability retention, the high molecular weight lignosulfonates performs far better than the low molecular weight ones. The former types also performs considerable better compared to a naphthalene polymer (i.e. sulfonated naphthalene formaldehyde), which serves as a reference in this work.

Generally, the yield value τ_o is found to increase with increasing time. Most often, it is also increasing with increasing temperature. However, there are exceptions to this: For mortar samples of ordinary Portland cement (OPC) using high molecular weight lignosulfonates, there is a clear decrease in yield value τ_o with increasing temperature. A possible explanation for this, is related to increased swelling and/or increased adsorption of the polymers to the surface of cement particles (see Figure 6.30, Page 138). This temperature effect does not occur for the case of fly ash cement (FAC) with the same lignosulfonate types. The reason for this may possible be related to that the swelling/adsorption properties of these polymers are overshadowed by a too large quantity of water present in the FAC-batches (see Section 6.4.4). For further details about these results see Section 6.5.3.

Part Two: Thixotropy (Measurements and Numerical Simulations)

To increasing the fundamental knowledge of rheological behavior of the cement based material, an investigations of thixotropic behavior was made on cement pastes mixed with either lignosulfonates or naphthalene. The thixotropic behavior is directly related to coagulation, dispersion and re-coagulation of the cement particles. In making the analysis, a modification is applied to the Hattori-Izumi theory, which is a theory about the bookkeeping of the number of reversible coagulated connections between the cement particles J_t. The modification consist, among other things, of include a fading memory to the analysis. That is, the cement paste is allowed to remember its recent past. By a combination of experimental results and numerical simulations, it is demonstrated that such memory term is very important. Also, two types of yield values had to be introduced into the simulation, namely τ_o and $\tilde{\tau}_o$. The former (τ_o) is related to the permanent coagulation state J_t^p of the cement particles, while the latter ($\tilde{\tau}_o$) is related to the reversible coagulation state J_t (see Figure 2.11, Page 27). The same type of relationship applies for the plastic viscosity μ and its thixotropic counterpart $\tilde{\mu}$. For further details about these results see Section 9.9.4.

Part Three: Errors of Different Kinds (Particle Migration)

Different types of errors that affect the measured rheological values have been considered. For the concrete batches of this thesis, the error of particle migration has been analyzed (Chapter 10). Realizing the nature of this error, some corrections are applied. However, with these corrections, one is only extracting the viscometric values of a "fat" concrete that surrounds the inner cylinder of the viscometer after

the particle migration is basically complete, and not of the concrete in the original homogenous state. In Section 6.3, a good relationship between the Abrams slump cone and the viscometer is established (see Figure 6.22, Page 123), indicating that the "fat" concrete simulates well the behavior of the concrete in a homogeneous state. For further details about these results see Sections 10.2.3, 10.3.8 and 6.3.

The errors resulting from plug flow (Section 3.5.3) and air entrainment (Section 5.5.3) are considered for the mortar measurements. For the given geometry and angular velocity of viscometer, the error generated from plug flow is demonstrated to be small when $\tau_o/\mu \leq 100\,\mathrm{s}^{-1}$. This condition applies in most cases in this thesis. When present, air instability (by air entrainment during (re)mixing) is shown to affect the plastic viscosity μ evolution to some significant degree. However, in such cases the yield value τ_o is more or less not affected.

Acknowledgments

I would like to express my gratitude to my main supervisor Professor Erik J. Sellevold for his personal commitment, interesting discussion and valuable advise. Erik has been continously encouraging and patient throughout my study. I will also express my gratitude to supervisors Professor Fridtjov Irgens and Sverre Smeplass for their valuable contributions to the thesis. Fridtjov has made an important contribution to the theoretical work of this thesis, while Sverre to the experimental work.

Special thanks to Dr. Kåre Reknes and Dr. Ólafur H. Wallevik for their valuable support and discussion. Kåre has been my contact person with Borregaard LignoTech and has made many valuable suggestions to the experimental work. Many thanks to Dr. Bernt O. Myrvold for further insight to lignosulfonate chemistry.

Furthermore, special thanks to my colleagues at the Department of Structural Engineering and associates at NORCEM, NORSTONE and SINTEF (Civil and Environmental Engineering, Cement and Concrete).

This work was made possible by the financial support from Borregaard LignoTech, Norway and the Research Council of Norway. Special thanks goes to Borregaard LignoTech for initiating this research project.

For the execution of the experimental program, special thanks to (in alphabetic order): Andreas Gurk, Bjørn Ingebrigtsen, Gøran Loraas, Hanna De Groote, Mark Benaets, Olav Haldorsen and Ove E. Loraas. Many thanks to other personals at M, K and P laboratories, for their fellowship and contribution to the experimental work.

I am also grateful to family and friends for their support, encouragement and patience during these last years.

Trondheim, February 2003
Jón Elvar Haraldsson Wallevik

Contents

Chapter 1

Introduction

1.1 Background and Short Review

Short Historical Background of Concrete [16, 86, 88]

Ever since civilizations first started to build, the human race have sought a material that bind stones into solid formed mass. The Romans mixed lime (i.e. burned limestone) with volcanic ash from Mount Vesuvius. This produced structures of remarkable durability. During the Middle Ages, the art of making hydraulic cement (cement which hardens when it comes in contact with water) became lost and it was not until the year of 1824 that the hydraulic cement (now commonly known as Portland cement) reappeared when it was patented by a Leeds builder named Joseph Aspdin. The name "Portland cement" is given originally due to the resemblance of the color and quality of the hardened cement to Portland stone (limestone quarried in Dorset).

Concrete and Rheology [88, 128]

After the rediscovery of cement, the concrete (essentially a mixture of Portland cement, water and aggregates) has become the most commonly used structural material in modern civilizations (see Figure 1.1). The quality of the concrete structure is of course dependent on the quality of each constituent that is used in the concrete mix. However, this is not the only controlling factor. The quality is also much dependent on the rheological properties of the fresh concrete during placement into the formwork. That is, the concrete must be able to flow into all corners of the mold or formwork to fill it completely. This is a process that might be made more difficult by the presence of awkward sections or congested reinforcement (see the right illustration of Figure 1.1). Tragic results often originate from concrete of unsuitable consistency, hardening into a honeycombed, non-homogenous mass. Therefore, one of the primary criteria for a good concrete structure is that the fresh concrete has satisfactory rheological properties during casting. With satisfactory properties, it is meant that the concrete can be placed into the mold or formwork without excessive effort, or sometimes without an effort at all. The latter type of concrete is known as self-compacting concrete. In addition, the concrete batch must always be such that it does not segregate during and after casting. This is the requirement of "stability". Unfortunately, such stability is not always achieved in reality at building sites.

Figure 1.1: Concrete has become the most commonly used structural material used in modern civilizations. To the left is a typical concrete structure, while center demonstrates a construction of oil-platform, made of concrete. To the right: The flow of concrete inside a formwork is often made more difficult by the presence of congested reinforcement.

Workability, Consistency, Flowability, Mobility and Pumpability

From the above, it is clear that the rheological properties must be well defined prior to casting. In practice, concrete that can be readily placed into a formwork is referred to as workable. This is a rather loose description of the rheological behavior of concrete and can also be very subjective, depending on type of formwork, type of concrete and the means of compaction available at building site [88]. Terms like workability[1], consistency[2], flowability, mobility and pumpability have been used to describe the rheological behavior of the fresh concrete. These terms are more reflected by personal viewpoint rather than of scientific precision [88, 128]. An interesting discussion about the subjectiveness of the term "workability", is given in a textbook by Tattersall and Banfill [128]. Other similar terms are also discussed there. The primary problem is that there is no guarantee that such a particular term means the same thing to different people. In fact, there has been a disagreement between different workers about the exact meaning of the term "workability" [87].

Even in ready-mixed plants, it is common for the final decision on water addition (to increase workability) to be made by the batcher-man on the basis of his own observation of the concrete behavior in the mixer and/or at discharge [128]. In some plants he is assisted in this task by the readings of a wattmeter or ammeter connected in the power supply line to the mixer, and by some empirical knowledge of the relationship between that reading and a certain slump of the concrete [128].

Rheological Measurements on Concrete

During the course of time, rheometers of different types and quality have been developed and used, to give some kind of rheological description of the fresh concrete. One of the most famous, oldest and currently most used empirical test is the so-called slump test. As is described in Section 4.4.2, it gives only a single value, namely the slump value S. The test was developed in the USA around 1910 [10]. It is believed that it was first used by Chapman [22] although in many countries the test apparatus is associated with Abrams [1, 10]. After this, other different empirical rheological

[1] One definition of the term "workability" is [ASTM C 125-93]: *A property, determining the effort required to manipulate a freshly mixed quantity of concrete with minimum loss of homogeneity.*

[2] One definition of the term "consistency" is [ASTM E24-58T]: *The resistance of a non-Newtonian material to deformation.*

tests have been developed like the flow/spread table test, which was developed in Germany in 1933 by Graf [37]. Other rheometers have followed since then, which have in common that they only provide a single rheological value. For example, the flow/spread table test gives a "flow diameter". Tattersall et al. [127, 128, 129] have criticized such workability tests on the grounds that they measure only one parameter. It is pointed out [127, 128, 129] that all empirical rheological tests are single-point tests, i.e. the result is a single number. One of the problem with such a number is that the same number may be produced by two concretes with quite different rheological characteristics. It is also mentioned that the empirical tests are very often operator-sensitive, in the sense that minor variations in the technique of carrying out the test, gives a different result. In the same literature [127, 128, 129] it is discussed the need for describing the rheological properties of fresh concrete in terms of fundamental physical quantities, not depending on the details of the apparatus with which they are measured. It is argued [127, 128, 129] that the concrete can, with good accuracy, be considered as Bingham fluid. In fact, then for the past 25 years, this has been demonstrated for fresh concrete, mortar and cement paste [28]. However, the Bingham model is not the only one that can been applied to cement based materials. For example, in [26, 31], it is concluded that fresh concrete corresponds to the Herschel-Bulkley model, rather than to the Bingham model. Also, Papo [94] gives a review of some of the models that have been applied to cement pastes. In Chapter 9 of this thesis, a new rheological model is represented and successfully used for cement pastes of thixotropic nature. This model is based on the Hattori-Izumi theory [42, 41] (see Section 2.4.2 about the Hattori-Izumi theory).

According to Tattersall and Banfill [128], a coaxial cylinders viscometer was not used for concrete until after about 1970. Motivated by the fact that such instrument had already been used for mortar and cement paste for quite a while before this, Tattersall made an attempt to apply a coaxial cylinders geometry to measure the rheological properties of fresh concrete [128]. Unfortunately, he was not successful. After this, in 1973, he introduced the use of a modified food mixer [125, 127, 128], to extract the Bingham parameters, namely the plastic viscosity[3] μ and the yield value τ_o. This configuration is known as the Mk I. A further development of the Mk I resulted in the famous Mk II and Mk III devices [128]. The Mk II is for highly workability concretes while the Mk III is for lower workability concretes. For all devices, empirical values G and H are measured. Through special calibration techniques [128], G and H are converted into fundamental physical quantities, namely the yield value τ_o and plastic viscosity μ. The Mk-systems are still being used and are continously going through some improvments as reported in different papers [28, 126, 129]. The Mk-systems are also known simply as the "two-point" and are commercially available. One of the newest Mk-system is shown in the left illustration of Figure 1.2.

In the beginning of the 1970's, trials were made by others [82, 140] using the coaxial cylinders system, which were reported to be somewhat more successful than the work by Tattersall. The inner and outer cylinders consisted of vanes, similar to what is shown in Figure 3.5 (Page 56). This was done to avoid slippage. In the late 1980's, a further improvement of the coaxial cylinders geometry was made in Norway [145, 147], which among other things consisted of that the bottom part of the inner cylinder did not measure torque (i.e. the "inner cylinder, lower unit" in Figure 3.4, Page 55). This was done to avoid the effect of shear stress τ generated from the lower part of the geometrical system used. This approach is validated in this thesis

[3]With Equation 3.3 (Page 53) the Bingham model is described.

by numerical simulations (see Sections 8.7 and 9.5). This viscometer is named the ConTec **BML** Viscometer 3, and is commonly known as the **BML** viscometer (see the right illustration of Figure 1.2). Ever since its first introduction, the ConTec **BML** Viscometer 3 has been continously going through improvements, in terms of more accurate instrumentation's and enhanced software (FreshWin) for controlling the viscometer. The viscometer is commercially available.

In 1996 a viscometer for fresh concrete, using parallel plates, was developed in France [50, 51] and is also commercially available. This viscometer is designated as BTRHEOM and is shown in the center illustration of Figure 1.2. A numerical simulation of this device is made in this thesis (see for example Figure 10.23, Page 257).

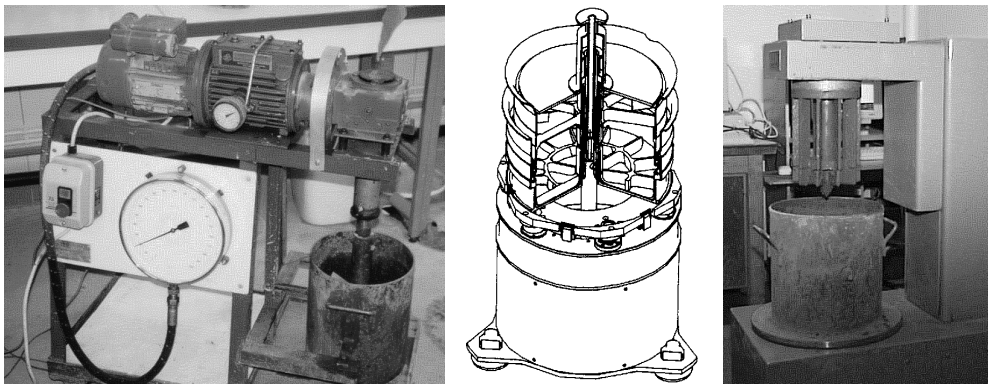


Figure 1.2: The Mk-system (to the left), the BTRHEOM (center) and the ConTec **BML** Viscometer 3 (to the right). See Figure 1.3 for comparison of rheological values.

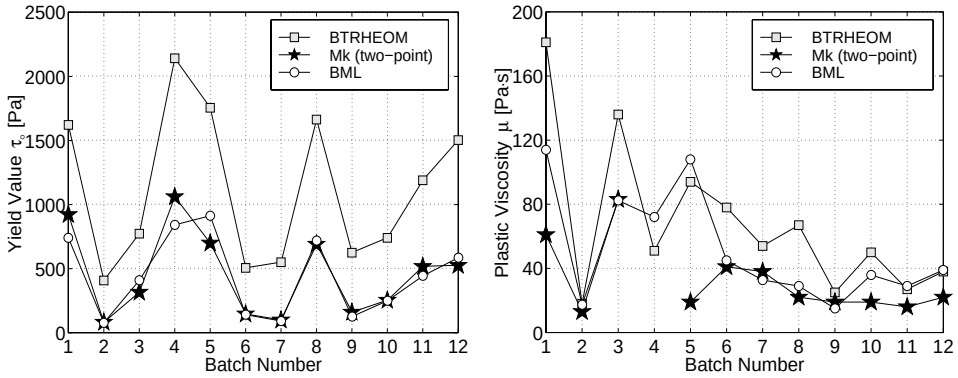


Figure 1.3: Comparison of rheological values measured by the Mk-system, ConTec **BML** Viscometer 3 and the BTRHEOM. The same batch is applied in each case. From [8].

The Mk-system, ConTec **BML** Viscometer 3 and the BTRHEOM all measure the rheological properties of fresh concrete in terms of fundamental physical quantities, namely in yield value τ_o and plastic viscosity μ. A comparison of these viscometers were made in France in October 2000, where the same concrete batch was simultaneously tested in the three devices [8]. Ideally, one should expect that the same set of rheological values (τ_o, μ) should be measured by all viscometers, whereas in reality this was not so. This is shown in Figure 1.3. Although the same changes in

rheological behavior are roughly measured from batch to batch, the absolute values of τ_o and μ differ somewhat, however, with the best agreement between the Mk-system and the ConTec **BML** Viscometer 3.

Rheological Measurements on Mortar and Cement Paste

Because rheological measurements on concrete requires large resources in terms of test material, labor and time, it is not practical to apply a complete laboratory test program to concrete. Rather, mortar and/or cement paste are used. This is done in the anticipation that the mortar and cement paste will simulate the rheological behavior of the concrete. By this approach, it is possible to create a larger and more complicated test program, to for example investigate the rheological effects of different admixture types. For such cases, different viscometers of smaller dimensions are usually used. In this thesis, the ConTec Viscometer 4 is used for such purpose. It is considerable smaller than the ConTec **BML** Viscometer 3 (see Figure 3.1, Page 51). The ConTec Viscometer 4 was introduced[4] in 1997 [146]. The use of coaxial cylinders geometry, when measuring the rheological properties of mortar and cement paste goes back to 1941, with the use of the Plastometer developed by Powers and Wiler [96, 97].

Mortar batches are not only mixed for academic purposes. They are also used at building site, as for example, bricklaying mortars, mortars for repair, patching or filling, whether premixed and packaged or mixed at the point of use; floor screeds and leveling compounds; decorative and waterproofing finishes [5]. Hence, the rheological properties of mortar can be more relevant than indicated in the previous paragraph.

Admixtures: Water-Reducers (P) and Superplasticizers (SP)

Almost always, some amount of chemical admixtures are added to the concrete batch during mixing. The reason for the use of such admixture[5] is that they are capable of imparting considerable physical and economical benefits to the concrete, mortar or cement paste. Traditionally, concrete has often contained minor addition of inorganic and organic materials. The use of naturally occuring hydraulic binders such as pozzolanas and lime was made by the Romans [16, 86, 88]. The use of blood as an air-entraining agent, as well as goat's milk and pig's fat to improve the workability was used in these early concretes [29]. Also, urine was applied to vary the rates of setting and hardening [29].

After the rediscovery of Portland cement in 1824, different admixtures for different purposes have been specifically developed for concrete, mortar and cement paste. A summary and discussion of the existing types is given in [29, 131]. In this thesis, only admixtures categorized as plasticizers (i.e. water-reducers) and superplasticizers are considered. These are lignosulfonates of various types and a sulfonated naphthalene formaldehyde. The purposes of using plasticizers[6] [89] (P) may be either of the following: **1**) to achieve a higher strength by decreasing the water/cement-ratio (i.e.

[4]Recently, additional viscometers have been introduced into the ConTec family. In the year of 2001, ConTec Viscometer 5 was released, and year later the ConTec Viscometer 6 was released (see also http://www.contec.is).

[5]Admixture implies addition at the mixing stage, while additive refers to a substance which is added at the cement manufacturing stage [89].

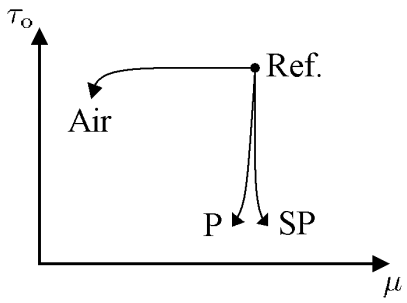
[6]The plasticizers used in this thesis are designated as **LMW Na**, **LMW Ca** and **LMWFS Ca**. The superplasticizers are however designated as **VHMW Na**, **HMW Na** and **HMW Ca** (and also **SNF**). For further readings about these admixtures, see Section 4.2.2.

w/c-ratio) at the same workability as an admixture-free mix; **2**) to maintain the same workability at lower cement content; **3**) to increase the workability so as to ease placing.

The superplasticizers (SP) are more effective type of water-reducers (i.e. plasticizers) and are also known as *high-range water-reducers* [89]. The dosage levels are usually higher than with conventional plasticizers since the undesirable side-effect of excessive retardation is considerably reduced [89, 105]. In this manner, they allow much greater water reduction than plasticizers, or alternatively allow extreme workability of the cement based material.

The earliest known published reference of plasticizers to increase workability, was made in 1932 [105]. During the mid 1930's to early 1940's, the use of lignosulfonates as plasticizers started [105]. Considerably later, or in the 1960's, superplasticizers for cement based material emerged [105]. These were the *sulfonated naphthalene formaldehyde* in 1963 (**SNF**) and the *sulfonated melamine formaldehyde* in 1964 (**SMF**) [29, 105]. Today, some special types of lignosulfonates are also categorized as superplasticizers. In this thesis, such types of lignosulfonates are tested and compared. In addition, **SNF** is also tested here as a reference. Many different plasticizers and superplasticizers are available. These are the **SNF**, **SMF** and lignosulfonates of different quality and types. Frequently, some of these are blended together and sold as a new product.

In his earlier work [144], Wallevik measured the relationships between the rheological parameters for concrete τ_o and μ and plasticizers (P) and superplasticizers (SP). The result is shown in the schematic figure to the right (reproduced from [144]): With increasing P or SP, the yield value τ_o reduces, while the plastic viscosity μ remains roughly unchanged. Also shown is the effect of increased air in the concrete. Within a certain range, increased air content mostly influences the plastic viscosity μ, while the yield value τ_o remains roughly unchanged. However, with sufficient large air content, the yield value is also affected.



Fresh Concrete, Mortar and Cement Paste as Particle Suspensions

In terms of rheology, it is natural to consider fresh concrete, mortar and cement paste as different types of suspensions. Other examples of suspensions are paint, printing inks, coal slurries and drilling muds [9]. Traditionally, a **particle suspension** consist of two phases, namely the **suspended particles** and the **matrix**.

As shown in the right illustration of Figure 1.4, fresh concrete consists of particles with a broad range of mass, dimension, shape and surface texture, suspended in a matrix (this applies also for mortar and cement paste). The distinction between matrix and suspended particles is a matter of choice, in contrast to the more traditional suspension of spheres submerged in a Newtonian liquid. For concrete, the matrix is defined here by pure convenience to be the $0-2\,\text{mm}$ mortar inside it (Chapters 4, 5, 6 and 10); i.e. the $0-2\,\text{mm}$ aggregates and cement particles are considered to only have a thickening effect on the water phase, as discussed in [9]. Such an approach is quite common. For example, Mørtsell [80, 81] treats the $0-0.125\,\text{mm}$ filler modified cement paste as matrix, instead of the pre-mentioned mortar. In Chapter 9, when

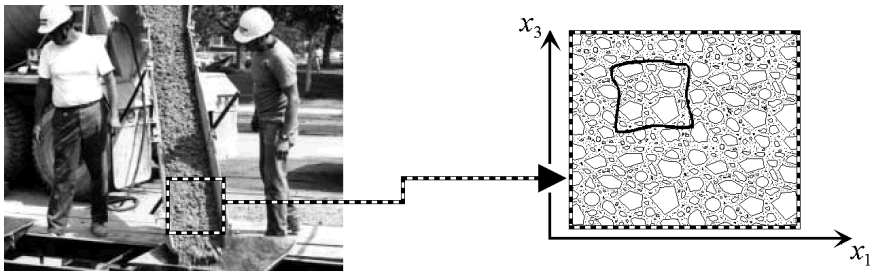


Figure 1.4: To the left: Casting of fresh concrete into a formwork [24]. To the right: Schematic presentation of the concrete. This particular illustration is also used in Figure 2.2 (Page 12) when underlying the basis for describing the concrete, mortar and cement paste as a pure continuum.

investigating the rheological properties of cement paste, its matrix is defined as the (free) water (see Footnote 19, Page 25 about the definition of free water).

Traditionally, when treating suspensions, one consider a shear viscosity function dependent on the phase volume Φ. For example, in 1905, Einstein showed that the shear viscosity of a very dilute suspension consisted[7] of $\eta = \eta_\circ(1 + 2.5\,\Phi)$ [9, 123]. The term $\eta_\circ$ is the shear viscosity of the matrix, while Φ is the phase volume (or solid volume fraction), defined as $\Phi = V_\mathrm{p}/(V_\mathrm{p} + V_\mathrm{m})$ [9]. The term V_p is the volume of the suspended particles and V_m is the volume of the matrix. Different types of shear viscosity functions η for different suspensions are presented in a textbook by Barnes et al. [9]. Generating a shear viscosity function η dependent on the phase volume Φ is not attempted in this thesis. However, such consideration has been recently made by de Larrard [25] in his new published textbook. As will be clear shortly, the objectives of this thesis is rather to investigate how the different lignosulfonate types changes the rheological behavior as a function of temperature and time, where the phase volume Φ (regardless of how defined) is the same for the two compared batches.

1.2 Objectives of this Thesis

There are three main objectives of this thesis, and it is consequently divided into three parts. The parts are independent in a certain sense, but in another sense dependent on each other. The ConTec **BML** Viscometer 3 is used to measure concrete, while the ConTec Viscometer 4 is for mortar and cement paste.

1.2.1 The First Objective: Effects of Lignosulfonates

This topic is the subject of **Chapters 2**, **3**, **4**, **5** and **6**. The objective is to determine how the different lignosulfonate types changes the rheological properties of the cement based material as a function of temperature and time. In particular:

1. Investigate the change in rheological behavior of mortar and concrete as a function of time, using the different types of lignosulfonates (see Section 4.2.2). For mortar, this investigation is made at three different temperatures, namely 5°C, 23°C and 38°C. The effects of a **SNF** product is also investigated for

[7] Einstein's equation neglects the effects of other particles. When particle-particle interactions are included, a higher order terms in Φ is generated [9, 123].

comparison. The results of the investigations are shown in Section 6.4 and is summarized in Section 6.5.3.

2. Investigate the rheological relationship between concrete and mortar. The mix design of the mortar is calculated from the concrete according to Section 4.3.2. The results of this investigation are shown in Section 6.3.2.

3. Investigate the relationship between rheological values generated by the ConTec **BML** Viscometer 3 with the values generated with the Abrams slump test (see Section 4.4.2 about the slump test). The results of this investigation are shown in Section 6.3.1.

1.2.2 The Second Objective: Thixotropy

This topic is the subject of **Chapters 2**, **3**, **7**, **8** and **9**. The objective is to identify some of the parameters affecting the shear viscosity η. The result of this investigation is helpful, when considering the rheological behavior of the cement based materials, presented in Chapter 6. In more exact terms, the second objective of this thesis is:

4. To investigate the thixotropic behavior of cement paste and in doing so, investigate if the Hattori-Izumi theory works. In Section 2.4.2, a description is made of the physics behind this theory. In the experiments, three different types of lignosulfonates are used. The **SNF** product is also used. The results of this investigation are summarized in Section 9.9.4.

1.2.3 The Third Objective: Particle Migration

This topic is the subject of **Chapters 7**, **8** and **10**. Migration of suspended particles from a region of high shear rate, to the region of low shear rate has been reported elsewhere [9, 70]. In this thesis, such a phenomenon is observed when conducting a rheological measurement on concrete. It is the gravel particles (here, the $2 - 16\,\text{mm}$ aggregates) that are moved from the region of high shear rate to the region of low shear rate. Hence, in this work, the process will be often referred to as gravel migration, but generally in the literature, such process is usually designated as particle migration [9, 70]. The results of this investigation are used in Chapters 5 and 6. More precisely, the third objective of this thesis is as listed below:

5. Investigate and analyze how much the phenomenon of gravel migration influences the viscometric values retrieved by the ConTec **BML** Viscometer 3. The results of this investigation are summarized in Section 10.2.3.

6. Other types of viscometers are investigated in relation to gravel migration. The purpose is to find a geometry that gives small potential for gravel migration. This investigation is based on numerical calculations alone. The results of this investigation are summarized in Section 10.3.8.

1.3 What is not Included in this Thesis

In Section 1.1, a short review was given on the rheology of cement based materials. The subject goes at least back to 1910 with the slump test. Since then, a large amount of discovery, suggestions, failures and improvements have been made. Obviously, a

review of all the work carried out would be a rather comprehensive task and the text of Section 1.1 only scrapes the surface of the most important issues in this field of science. Instead of making a comprehensive review, the main effort has been focused on the three basic objectives of the thesis, given in Section 1.2. Basically, one could state that the work done here is based on the previous work done by Tattersall [127, 128, 129], Hattori and Izumi [42, 41]. Tattersall introduced the idea of using the Bingham model for the cement based material and retrieving the corresponding Bingham parameters by a viscometer. Hattori and Izumi introduced the idea of consider thixotropy as a result of coagulation and dispersion of the cement particles (in Section 9.1, a short review about the thixotropy is given). Obviously, the work of many others is also involved in this thesis and appropriate references are given.

Chapter 2

Description of Fluid

2.1 Introduction

In this thesis, the fluid state (or more precisely, the viscoplastic state) of the fresh concrete, mortar and cement paste will be the main topic. Consequently, it is only natural to begin with the question *what is a fluid*. When considering a large collection of rock and ice fragments, as in the rings of Saturn[1], or a large collection of sea ice floes in the Icelandic waters (see Figure 2.1), one might have difficulties in accepting those two systems as fluids. Of course, in their isolated state, a single rock or ice fragment, or a single sea ice floe, does not represent such state. But with a large collection of these solid particles, those two separated systems can be presented as two different types of fluid. They are classified and characterized by their potential solid ice/rock and sea ice interactions through parameters called viscosity.

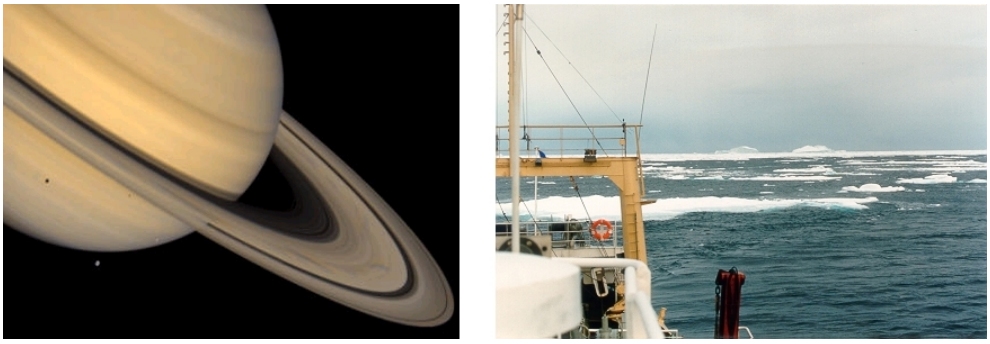


Figure 2.1: Two different types of fluid. To the left: The rings of Saturn, composed of rock and ice fragments [119]. To the right: Sea ice floes in the Icelandic waters [57].

With the introduction shown in the following section, it will be clear that the fluid approach consist of calculating the motion of a large number of solid particles, without going into the detailed motion of every single one of them. For example, in the case of fresh concrete, where the solid particles consist of a broad range in mass, dimension, shape and surface texture (see Figure 2.2), the largest solid particles (i.e.

[1]The rings of Saturn are mostly made of ice fragments, ranging from few centimeters to few meters across, but there are also some traces of silicate and carbon minerals, indicating that rock fragments are also present [3].

the largest gravel particles) are treated in the exactly the same manner as the smallest ones (i.e. the water molecules). In this perception, both the suspended particles and the matrix are treated in exactly the same manner, namely together as a group of solid particles. No distinction is made between the two phases.

2.2 The Governing Equation

When dealing with the motion of a (enormously) large collection of solid particles, the concept of a particle must be redefined. The reason is that it is rather difficult, if not impossible, to gain a solution for multi particle system by looking at every single **solid particle**[2] as the *working* particle of the system. That is, by using Newton's 2nd law ($d\mathbf{p}_\mathrm{I}/dt = \mathbf{F}_\mathrm{I} + m_\mathrm{I}\,\mathbf{g}$) on every such particle, some set of problems arises, which are explained in Appendix B.2.1. Instead of directly using the solid particle, an alternative approach is to create a new kind of particle, namely the The Continuum Particle[3] (CP). In doing so, the pre-mentioned large collection of solid particles has now the designation: The Continuum. The continuum and its boundary will be designated here with the symbols Ω and $\partial\Omega$, receptively. The CP is composed of large amount of solid particles (see Figure 2.2). For example, in the case of pure water continuum, the CP consists of vast amount of water molecules. As for homogeneous[4] fresh concrete, every CP must consist of collection of aggregates, cement grains, water molecules and so forth, as shown in Figure 2.2 to the left.

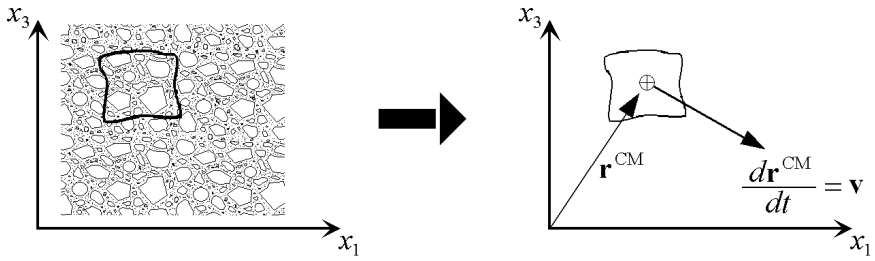


Figure 2.2: To the left: A cross section of a particle suspension, like of the fresh concrete, the rings of Saturn or of the sea ice floes. A CP is composed of all the solid particles that are confined within the CP boundary, shown with the black solid line. To the right: Graphical illustration for Equation 2.7. Illustration from the left to the right, demonstrates the transition from the solid particle-approach to the CP-approach. [$\oplus$ designates the center of mass of a CP].

The CP is defined here as such that it always consists of the same magnitude of mass δm for the time domain of calculation. In mathematical terms this means:

$$\delta m = \text{constant} \tag{2.1}$$

As shown in the left illustration of Figure 2.2, the CP is composed of a large collection of solid particles. In the derivations that now follows, it is assumed that for a

[2]For example, looking at every single aggregate, cement grain, water molecule and so forth.

[3]The term *Continuum Particle* is not the only one used in the literature of fluid and continuum mechanics. To mention few, *Material Point* [43, 53, 93], *Fluid Particle* [93] and *Fluid Parcel* [20] are terms also used in various textbooks.

[4]With homogeneous suspension, it is meant that all constituents are (more or less) equally distributed throughout the continuum. If bleeding or segregation occurs, such condition does not apply.

given period of time, their total number is N. Each such solid particle, changes their momentum according to Newton's 2nd law, as shown with Equation 2.2.

$$\frac{d\mathbf{p}_\mathrm{I}}{dt} = \mathbf{F}_\mathrm{I} + m_\mathrm{I}\,\mathbf{g} \quad \forall \ \mathrm{I} \in [1, \mathrm{N}] \tag{2.2}$$

The term $\mathbf{p}_\mathrm{I} = m_\mathrm{I}\mathbf{v}_\mathrm{I}$ is the momentum of the solid particle number I. $\mathbf{v}_\mathrm{I} = d\mathbf{r}_\mathrm{I}/dt$, $\mathbf{r}_\mathrm{I}$ and m_I represents its velocity, vector position and mass, respectively. Finally, $\mathbf{F}_\mathrm{I}$ is the sum of external surface forces applied to this solid particle, from its surrounding solid particles and $m_\mathrm{I}\,\mathbf{g}$ is the body force (or gravity force) applied to it. The summation of Equation 2.2 for all I from 1 to N, gives the equation of motion for the specific CP in question. This equation is shown below:

$$\frac{d}{dt}\sum_{\mathrm{I}=1}^{\mathrm{N}} \mathbf{p}_\mathrm{I} = \sum_{\mathrm{I}=1}^{\mathrm{N}} (\mathbf{F}_\mathrm{I} + m_\mathrm{I}\,\mathbf{g}) = \sum_{\mathrm{I}=1}^{\mathrm{N}} \mathbf{F}_\mathrm{I} + \delta m\,\mathbf{g} \tag{2.3}$$

where $\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I} \equiv \delta m$. As demonstrated with the simple example in Figure B.1 (Page 378), the Newton's 3rd law, the law of action and reaction, always governs between the solid particles located at the interior of the CP. As such, the internal surface forces (i.e. inside the CP) will summarize to zero. However the external surface forces applied to solid particles located at the boundary of this CP, will remain. This result is fundamental in the sense that it applies regardless of the type of forces that interact between the solid particles. [The type of forces that could be involved are for example the mechanical collisions forces and/or forces of attractive/repulsive potential energy]. With this, one can rewrite Equation 2.3 as shown with Equation 2.4.

$$\frac{d}{dt}\sum_{\mathrm{I}=1}^{\mathrm{N}} \mathbf{p}_\mathrm{I} = \mathbf{F} + \delta m\,\mathbf{g} \tag{2.4}$$

As indicated above, the force $\mathbf{F}$ represents the sum of external surface forces applied to the CP from its surroundings. The term $\delta m\,\mathbf{g}$ is the body force applied to the CP.

Multiplying the expression "$1 = \sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I} / \sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I}$" with the left side of Equation 2.4, followed by some[5] rearrangement, results in Equation 2.5.

$$\left(\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I}\right)\left[\frac{d}{dt}\left(\frac{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I}\mathbf{v}_\mathrm{I}}{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I}}\right)\right] = \mathbf{F} + \delta m\,\mathbf{g} \tag{2.5}$$

From the previous definition, the above can equally be rewritten as follows:

$$\delta m\frac{d\mathbf{v}}{dt} = \mathbf{F} + \delta m\,\mathbf{g} \quad \text{where} \quad \mathbf{v} = \frac{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I}\mathbf{v}_\mathrm{I}}{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I}} \tag{2.6}$$

One can see that when considering the velocity $\mathbf{v}$ of a CP, it is meant the mass average velocity of all the solid particles composing the specific CP. Hence, any random and spontaneous velocity contributions from the individual solid particles are summarized out[6] in the averaging and only the relevant smooth motion of the continuum will

[5] $\sum_{\mathrm{I}=1}^{\mathrm{N}} m_\mathrm{I} \equiv \delta m =$ constant, c.f. Equation 2.1.

[6] When going from the step of Equation 2.2 to Equation 2.4, another equation is also implicitly formed in the background. This equation deals only with the random part of the velocity $\mathbf{v}_\mathrm{I}$, namely the $[\mathbf{v}_\mathrm{I} - \mathbf{v}]$ part (see Footnote 3, Page 378). This equation is generally known as the internal energy equation and is in the form of $\rho\,du/dt = \rho\,c\,dT/dt = \eta\,\dot{\gamma}^2$ (see Equation B.40 on Page 391) where $u = c\,T \propto \sum_{\mathrm{I}=1}^{\mathrm{N}} \frac{1}{2} m_\mathrm{I}\,(\mathbf{v}_\mathrm{I} - \mathbf{v})^2/\delta m$.

remain. The same consideration goes for the external surface force $\mathbf{F}$: Any random and spontaneous force contributions from the individual solid particles, are summarized out in the calculation of $\sum_{\mathrm{I}=1}^{\mathrm{N}} \mathbf{F}_{\mathrm{I}}$. Hence, only the relevant smooth external surface force $\mathbf{F} = \mathbf{F}(\dots)$ will remain (namely $\mathbf{F} = \delta V \, \nabla \cdot \boldsymbol{\sigma}$, c.f. Equation B.10 on Page 381).

The velocity $\mathbf{v}$ represents the velocity of the center of mass (CM) for the CP. This is apparent from the definition of the center of mass (see also Figure 2.2):

$$\mathbf{r}^{\mathrm{CM}} = \frac{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_{\mathrm{I}} \mathbf{r}_{\mathrm{I}}}{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_{\mathrm{I}}} \quad \Rightarrow \quad \frac{d\mathbf{r}^{\mathrm{CM}}}{dt} = \frac{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_{\mathrm{I}} \frac{d\mathbf{r}_{\mathrm{I}}}{dt}}{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_{\mathrm{I}}} = \frac{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_{\mathrm{I}} \mathbf{v}_{\mathrm{I}}}{\sum_{\mathrm{I}=1}^{\mathrm{N}} m_{\mathrm{I}}} = \mathbf{v} \tag{2.7}$$

The vector $\mathbf{r}^{\mathrm{CM}}$ designates the CM vector position of the CP. Now, instead of labeling each and every CP with, for example, a specific number, color or whatever, it is customary to mark them with their initial coordinates $\mathbf{X}$ [74]. Hence, the CM-position of a CP that has the label $\mathbf{X}$, is defined by Equation 2.8.

$$\mathbf{r}^{\mathrm{CM}} = \mathbf{x} = \mathbf{x}(\mathbf{X}, t) = x_{\mathrm{i}}(X_{\mathrm{j}}, t)\mathbf{i}_{\mathrm{i}} \tag{2.8}$$

To repeat, the above equation inform about the CM-position $\mathbf{x}$ of a CP at the time $t \geq 0$, which had the CM-position $\mathbf{X}$ at time $t = 0$. Equation 2.8 is referred to as the **Lagrange formulation**, since the initial position $\mathbf{X}$ of the CP acts now as the independent variable. In mathematical terms, $\mathbf{X}$ is defined by [74]:

$$\mathbf{X} = \mathbf{X}(\mathbf{x}, t) = X_{\mathrm{i}}(x_{\mathrm{j}}, t)\mathbf{i}_{\mathrm{i}} \tag{2.9}$$

The above equation can be viewed as one which provides a tracing of the CP from its current position $\mathbf{x}$ to its original position $\mathbf{X}$ [74]. Equation 2.9 is referred to as the **Eulerian formulation**, since the current position[7] of the CP acts now as the independent variable.

With the above labeling method in mind, let $\mathbf{v}(\mathbf{X}, t)$ designate the smooth CM-velocity of a CP that is labeled with its initial coordinates $\mathbf{X}$ (that is, with: $\mathbf{X}$ = "a specific constant"). Furthermore, let $\delta m(\mathbf{X})$ designate its mass and $\mathbf{F}(\mathbf{X}, t)$ the sum of external surface forces applied to it, from its surroundings. With these definitions, then from Equation 2.6, the governing equation for this particular CP becomes:

$$\delta m(\mathbf{X}) \frac{d\mathbf{v}(\mathbf{X}, t)}{dt} = \mathbf{F}(\mathbf{X}, t) + \delta m(\mathbf{X})\, \mathbf{g} \qquad \mathbf{X} = \text{"a specific constant"} \tag{2.10}$$

The above equation can equally apply for another CP, inside the continuum (that is, with: $\mathbf{X}$ = "a different constant"). In fact, with this property, Equation 2.10 can be used on any CP inside the continuum (that is, with: $\mathbf{X} \in \Omega$). Hence, all the governing equations of every CP, can be compressed so to speak, into only one equation:

$$\delta m(\mathbf{X}) \frac{d\mathbf{v}(\mathbf{X}, t)}{dt} = \mathbf{F}(\mathbf{X}, t) + \delta m(\mathbf{X})\, \mathbf{g} \qquad \forall \quad \mathbf{X} \in \Omega \tag{2.11}$$

With the above equation, then instead of working with multiple governing equations (one governing equation for each and every CP), one is now working with multiple coordinates $\mathbf{X}$ (one coordinates $\mathbf{X}$ for each and every CP) and with only one governing equation.

[7] An example of the Lagrangian description of the CM-position for a specific CP can be given by the following equations: $x_1(\mathbf{X}, t) = X_1 + X_2\,(e^t - 1)$; $x_2(\mathbf{X}, t) = X_1\,(e^{-t} - 1) + X_2$ and $x_3(\mathbf{X}, t) = X_3$. Accordingly, its (inverse) Eulerian position is then given by: $X_1(\mathbf{x}, t) = f(t)\,(-x_1 + x_2\,(e^t - 1))$; $X_2(\mathbf{x}, t) = f(t)\,(x_1\,(e^{-t} - 1) - x_2)$ and $X_3(\mathbf{x}, t) = x_3$ where $f(t) = 1/(1 - e^t - e^{-t})$ [74].

With Equation 2.9, one can transform any variable from Lagrangian description to the Eulerian description [74]. For example in case of the velocity in Equation 2.11, the following can be applied: $\mathbf{v}(\mathbf{X},t) = \mathbf{v}(\mathbf{X}(\mathbf{x},t),t) = \mathbf{v}(\mathbf{x},t)$. The mathematical functional form of the velocity term is generally not the same before and after such transformation. This is shown with Footnote 7. To emphasize this difference, a subscript is sometimes added: $\mathbf{v}_{\mathrm{L}}(\mathbf{X},t) = \mathbf{v}_{\mathrm{L}}(\mathbf{X}(\mathbf{x},t),t) = \mathbf{v}_{\mathrm{E}}(\mathbf{x},t)$, but this is redundant, since the difference is automatically understood by looking at the independent variable of the function in question. Now, after applying the same procedure to the rest of the variables in Equation 2.11, the following is produced:

$$\delta m(\mathbf{x},t)\frac{d\mathbf{v}(\mathbf{x},t)}{dt} = \mathbf{F}(\mathbf{x},t) + \delta m(\mathbf{x},t)\,\mathbf{g} \qquad \forall \quad \mathbf{x}(\mathbf{X},t) \in \Omega \tag{2.12}$$

This is the governing equation for a CP (labeled with $\mathbf{X}$) that is passing through the position $\mathbf{x} = \mathbf{x}(\mathbf{X},t)$ at the time t. At a later time $t + \Delta t$, the coordinates $\mathbf{x}$ is occupied by another CP (i.e. a different $\mathbf{X}$), which could have a different amount of mass $\delta m(\mathbf{X})$. Hence the time dependency in the mass function $m(\mathbf{x},t)$.

Dividing Equation 2.12 with the volume[8] $\delta V(\mathbf{X},t) = \delta V(\mathbf{X}(\mathbf{x},t),t) = \delta V(\mathbf{x},t)$ of a CP that is passing through the coordinates $\mathbf{x}$ at the time t, gives Equation 2.13.

$$\frac{\delta m(\mathbf{x},t)}{\delta V(\mathbf{x},t)}\frac{d\mathbf{v}(\mathbf{x},t)}{dt} = \frac{\mathbf{F}(\mathbf{x},t)}{\delta V(\mathbf{x},t)} + \frac{\delta m(\mathbf{x},t)}{\delta V(\mathbf{x},t)}\,\mathbf{g} \tag{2.13}$$

The mass of the CP divided by its volume is called **density** and is designated with ρ. Equation 2.14 defines density in both Lagrangian and Eulerian formulation.

$$\rho(\mathbf{X},t) \equiv \frac{\delta m(\mathbf{X})}{\delta V(\mathbf{X},t)} = \frac{\delta m(\mathbf{x},t)}{\delta V(\mathbf{x},t)} \equiv \rho(\mathbf{x},t) \tag{2.14}$$

Using the general chain rule of calculus [30] on the acceleration term in Equation 2.13, produces the following equation:

$$\frac{d\mathbf{v}(\mathbf{x},t)}{dt} = \left.\frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial t}\right|_{\mathbf{x}}\frac{dt}{dt} + \left.\frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial \mathbf{x}}\right|_{t} \cdot \frac{d\mathbf{x}(\mathbf{X},t)}{dt} \tag{2.15}$$

where $\partial\mathbf{v}/\partial\mathbf{x} \equiv \nabla\mathbf{v}$. The last part in the above equation $d\mathbf{x}(\mathbf{X},t)/dt$, is the velocity of the CP in Lagrangian description $\mathbf{v}(\mathbf{X},t)$, which can be transformed to Eulerian description: $\mathbf{v}(\mathbf{X},t) = \mathbf{v}(\mathbf{X}(\mathbf{x},t),t) = \mathbf{v}(\mathbf{x},t)$. Now, defining $\mathbf{f}(\mathbf{x},t) \equiv \mathbf{F}(\mathbf{x},t)/\delta V(\mathbf{x},t)$ as the sum of external surface forces per unit volume[9] and thereupon combining Equations 2.13, 2.14 and 2.15, the following is generated:

$$\rho(\mathbf{x},t)\left(\frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial t} + \mathbf{v}(\mathbf{x},t)\cdot\nabla\mathbf{v}(\mathbf{x},t)\right) = \mathbf{f}(\mathbf{x},t) + \rho(\mathbf{x},t)\,\mathbf{g} \tag{2.16}$$

This is the governing equation for a CP (labeled with $\mathbf{X}$) that is passing[10] through the position $\mathbf{x} = \mathbf{x}(\mathbf{X},t)$ at the time t. As such, it is important to bear in mind, that

[8] The time dependence in the function $\delta V(\mathbf{X},t)$ is only factual if the continuum (or equally, all the CPs that composes the continuum) is compressible, for example what applies for gases.

[9] In simplifying text, $\mathbf{f}$ will also be frequently referred as the sum of external surface forces.

[10] With the overall steps done from Equation 2.10 to Equation 2.16, it is clear that the operator $d/dt = \partial/\partial t + \mathbf{v}\cdot\nabla$ used in Equation 2.16, expresses the *time derivative following a CP*, which is labeled with its initial coordinates $\mathbf{X}$. This operator have been given different names like *substantial derivative*, *material derivative* [36, 48], *total derivative* [20, 36, 48], *convective derivative* [11] and *Eulerian derivative* [36]. In this thesis simply the term *time derivative* (contrary to the term *partial time derivative* $\partial/\partial t$) will suffice for this operator.

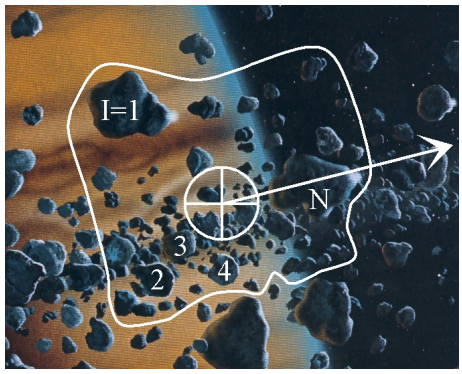
the coordinates $\mathbf{x}$ is the current CM-position $\mathbf{x}(\mathbf{X}, t)$ of a specific CP, labeled with $\mathbf{X}$. When the independent variables in the above equation are omitted, this should be automatically understood.

In Appendix B.3, the sum of external surface forces applied to the CP from its surroundings is calculated as $\mathbf{f} = \mathbf{F}/\delta V = \nabla \cdot \boldsymbol{\sigma}$ (see Equation B.10, Page 381). Putting this result in Equation 2.16, gives the following:

$$\rho(\mathbf{x}, t) \left(\frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \nabla \mathbf{v}(\mathbf{x}, t) \right) = \nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) + \rho(\mathbf{x}, t)\ \mathbf{g} \tag{2.17}$$

To conclude, Equations 2.17 and 2.3 are two different representations of the same principle, namely the Newton's 2nd law applied on a large collection of solid particles:

$$\frac{d}{dt} \sum_{\mathrm{I}=1}^{\mathrm{N}} \mathbf{p}_{\mathrm{I}} = \sum_{\mathrm{I}=1}^{\mathrm{N}} (\mathbf{F}_{\mathrm{I}} + m_{\mathrm{I}}\, \mathbf{g}) \quad \Leftrightarrow \quad \rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \nabla \cdot \boldsymbol{\sigma} + \rho\ \mathbf{g} \tag{2.18}$$



An example of a large collection of solid particles would be the rings of Saturn, consisting of a solid ice and rock fragments. Their dimensions range from few centimeters to few meters across [3]. Utilization of Equation 2.17 on the rings is done in Appendix B.5. Using this equation on these set of solid particles, is done to highlight that the fluid approach is in essence the theory of collective motion of a very large number of solid particles (see figure to the right[11]), without going into the detailed motion of every single one of them. The largest solid particles are treated in the exactly the same manner as the smallest ones. Equation 2.17 can equally be applied to other "bizarre" fluids, like the sea ice floes in the Icelandic waters (see Figure 2.3) or to the fresh concrete flowing inside a formwork (see Figure 3.2, Page 54) or inside a viscometer (see Figure 3.8, Page 62).

2.3 The Constitutive Equation

For many fluids, the constitutive equation is represented as $\boldsymbol{\sigma} = -p\,\mathbf{I} + \mathbf{T}$ [9], where the second order tensor $\mathbf{T} = T_{\mathrm{ij}}\mathbf{i}_{\mathrm{i}}\mathbf{i}_{\mathrm{j}}$ is known as the **extra stress tensor** and p is the pressure. The term $\mathbf{I}$, is known as the **unit dyadic** and its index equivalence is the **Kronecker delta**, written as δ_{ij} where $\delta_{\mathrm{ij}} = 1$ if $\mathrm{i} = \mathrm{j}$ and $\delta_{\mathrm{ij}} = 0$ if $\mathrm{i} \neq \mathrm{j}$ [53, 74]. In index notation, the tensor $\boldsymbol{\sigma}$ is written as: $\sigma_{\mathrm{ij}} = -p\delta_{\mathrm{ij}} + T_{\mathrm{ij}}$. According to customary understanding, σ_{ij} designates a stress in j-direction on a plane that has a normal unit vector pointing in i-direction [9, 53, 72, 74]. Furthermore, it can be shown that this tensor is symmetric: $\sigma_{\mathrm{ij}} = \sigma_{\mathrm{ji}}$ [72, 74]. The same considerations applies for T_{ij}. In Section 2.4.1, this tensor is associated with the exchange of momentum between solid particles of the continuum.

There is a branch in rheology that consists of determining the function form of the extra stress tensor $\mathbf{T}$ for the different types of fluid. An English introductory text

[11] This illustration is also shown in Appendix B.5 in its original state.

about the subject can be seen in a textbook after Barnes et al. [9]. For example, the tensor $\mathbf{T}$ for linear viscoelastic fluid is written in the form [9, 55]:

$$\mathbf{T}(\mathbf{x},t) = 2\int_{-\infty}^{t} \phi(t-t')\ \dot{\boldsymbol{\varepsilon}}(\mathbf{x},t')\ dt' \tag{2.19}$$

The variable ϕ is the relaxation function, t is the present time and t' is the earlier time. The tensor $\dot{\boldsymbol{\varepsilon}} = \dot{\varepsilon}_{\mathrm{ij}}\,\mathbf{i}_{\mathrm{i}}\mathbf{i}_{\mathrm{j}}$ is called the **strain rate tensor** and is given by:

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}}\right) = \frac{1}{2}\left(\frac{\partial v_{\mathrm{i}}}{\partial x_{\mathrm{j}}} + \frac{\partial v_{\mathrm{j}}}{\partial x_{\mathrm{i}}}\right)\mathbf{i}_{\mathrm{i}}\mathbf{i}_{\mathrm{j}} \tag{2.20}$$

In the last part of the above, a so-called *indicial notation* [74, 72] in Cartesian coordinate system, is used. The velocity gradient tensor $\nabla\mathbf{v} = [\partial v_{\mathrm{i}}/\partial x_{\mathrm{j}}]\mathbf{i}_{\mathrm{i}}\mathbf{i}_{\mathrm{j}}$ can be looked upon as a comparison of velocities between CPs, placed around the CP in question. For example, $\partial v_1/\partial x_3$ can be looked at as a comparison of the velocity component in the $\mathbf{i}_1$-direction, between two CPs, placed above and below the CP in question. They are separated by the distance $2\delta x_3$, relative to their CM as shown with Equation 2.21 and with the left illustration of Figure B.4 (Page 383).

$$\frac{\partial v_1}{\partial x_3} = \lim_{\Delta x_3 \to \delta x_3} \frac{v_1(x_1,x_2,x_3+\Delta x_3,t) - v_1(x_1,x_2,x_3-\Delta x_3,t)}{2\,\Delta x_3} \tag{2.21}$$

Another example of a constitutive equation is given with Equation 2.22. This equation have been used when predicting the flow of the two dimensional sea ice continuum [45, 46]. There the corresponding CP consists of large collection of sea ice floes[12], where each of them can have a horizontal dimension of several to tens of kilometers. Figure 2.3 demonstrate results from a numerical simulation when predicting the sea ice movement on Icelandic waters according to Equations 2.17 and 2.22 [142].

$$\boldsymbol{\sigma}(\mathbf{x},t) = -\frac{P(\mathbf{x},t)}{2}\,\mathbf{I} + \kappa\,\mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t)\right)\mathbf{I} + 2\,\eta\left[\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t) - \frac{\mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t)\right)}{2}\,\mathbf{I}\right] \tag{2.22}$$

The term $P/2$ is rather complex to be described in a short introductory text, but a short explanation would be that it plays a similar role as pressure p in more casual system of fluid flow. The terms $\kappa = \kappa(P,\dot{\boldsymbol{\varepsilon}})$ and $\eta = \eta(P,\dot{\boldsymbol{\varepsilon}})$ are the bulk viscosity and the shear viscosity, respectively.

The constitutive equation that is successfully used in this thesis (and elsewhere), has the following functional form:

$$\boldsymbol{\sigma}(\mathbf{x},t) = -p(\mathbf{x},t)\mathbf{I} + \mathbf{T}(\mathbf{x},t) \quad \wedge \quad \mathbf{T}(\mathbf{x},t) = 2\,\eta(\mathbf{x},t)\,\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t) \tag{2.23}$$

where p is the mechanical pressure[13] and η is the shear viscosity. As an example of a shear viscosity function that could be used in Equation 2.23 is the one for the Bingham model: $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$, where τ_{o} and μ are known as the **yield value** [128] (or

[12] This field of theory does not include the icebergs. Ice floes are generated on water and can have maximum thickness of several meters, but icebergs are generated on land as glaciers and later transported to the sea. The latter can have a thickness of several hundreds meters [124].

[13] In this work it will be assumed that the density of the fresh concrete, mortar and cement paste is a constant and not a thermodynamic variable. This assumption can be made since acoustic phenomenon is not of interests here. The mechanical pressure is defined by the average of normal stresses: $p_{\mathrm{m}} \equiv -\mathrm{tr}(\boldsymbol{\sigma})/3$ [93]. With Equation 2.23 and the pre-mentioned incompressibility ($d\rho/dt = -\rho\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}) = 0$), then $p_{\mathrm{m}} = -[\mathrm{tr}(-p\,\mathbf{I}) + 2\,\eta\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}})]/3 = p$.

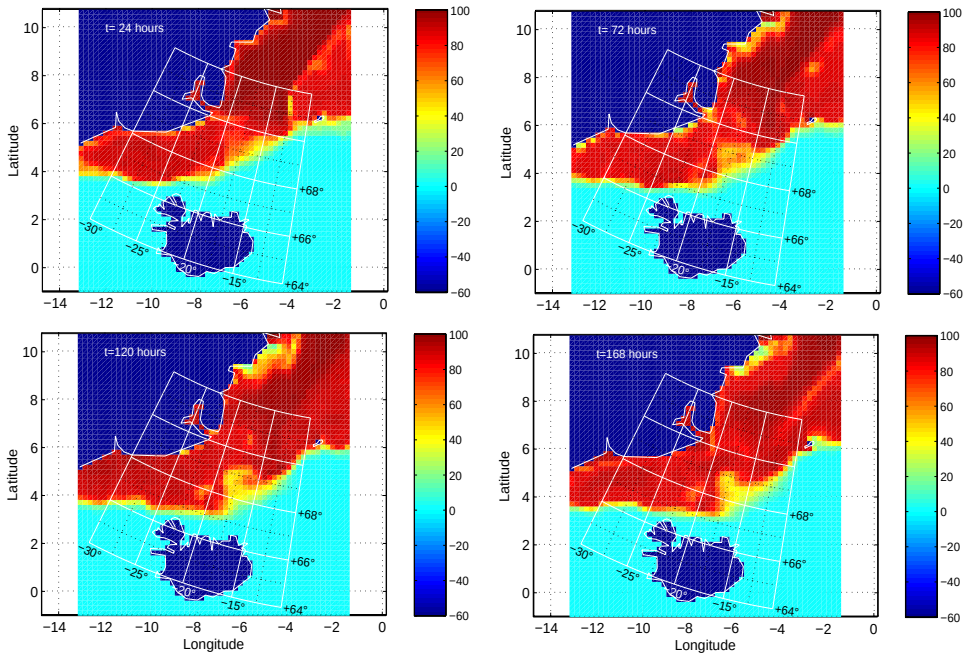


Figure 2.3: Total ice compactness in percentages on Icelandic waters. Initial condition is taken from the 18^{th} of January 1997 at $12.00\,\text{GMT}$ (0 hours). From left to right and from top to bottom are simulation results at 24, 72, 120 and 160 hours [142].

yield stress [9]) and the **plastic viscosity** [9, 128], respectively. Other types of shear viscosity function η might apply as well, depending on nature of the fluid material in question. The term $\dot{\gamma}$ is known as the shear rate and is a function of the strain rate tensor as shown with Equation 2.24 [55, 66].

$$\dot{\gamma} = \sqrt{2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}} = \sqrt{2\,\dot{\varepsilon}_{\text{ij}}\dot{\varepsilon}_{\text{ij}}} \tag{2.24}$$

Figures 2.4 and 2.5 demonstrates a result from a numerical simulation of a Bingham fluid inside the ConTec **BML** Viscometer 3 by using Equations 2.17 and 2.23 (see Chapter 8). Both illustrations are from the same calculation. The former figure demonstrates the velocity profile[14] $\mathbf{v} = v_\theta\,\mathbf{i}_\theta$, while the second one shows the profile[15] of von Mises shear stress $\tau = \sqrt{(\mathbf{T} : \mathbf{T})/2}$ (see Section 3.2). The angular velocity of the outer cylinder for this calculation is $\omega_{\text{o}} = 3\,\text{rad/s}$ and the viscometric values used, are $\tau_{\text{o}} = 1200\,\text{Pa}$ and $\mu = 120\,\text{Pa}\cdot\text{s}$. When considering a particle suspension like of concrete with maximum aggregate size of $D_{\max} = 16\,\text{mm}$, the smooth velocity profile shown in the first figure, does not represent the velocity for the individual solid particles (i.e. of the individual aggregate, cement grain or water molecule). This is because the latter type of motion is on a scale below what can be provided by Equation 2.17. The velocity profile of the individual solid particles is much more discontinuous and random than presented in this figure. Instead, Figure 2.4 shows

[14] The isolines start at $v_\theta = 5\,\text{cm/s}$ near the inner cylinder and with equal increments of $5\,\text{cm/s}$, ends at $40\,\text{cm/s}$ near the outer cylinder.

[15] The isolines start at $\tau = 300\,\text{Pa}$ near the outer cylinder and with equal increments of $300\,\text{Pa}$, ends at $4800\,\text{Pa}$ around the corner of inner cylinder.

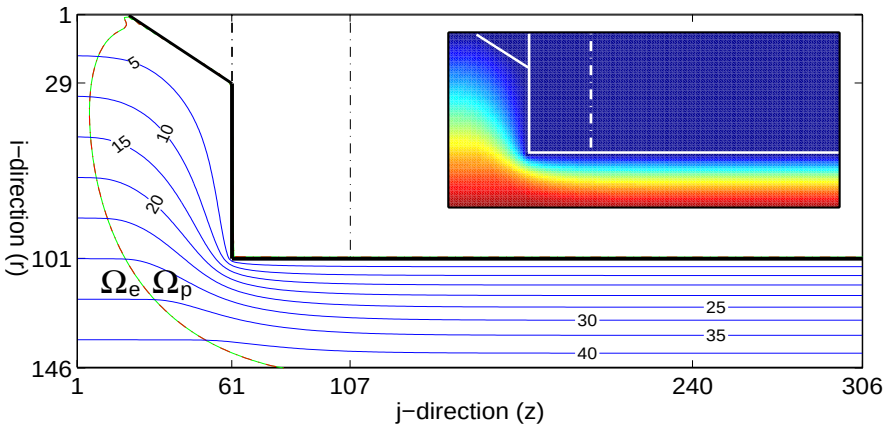


Figure 2.4: Velocity profile $\mathbf{v} = v_\theta\,\mathbf{i}_\theta$ (in cm/s) inside the ConTec **BML** Viscometer 3 when using the Bingham model $\eta = \mu + \tau_o/\dot{\gamma}$, with $\tau_o = 1200\,\mathrm{Pa}$, $\mu = 120\,\mathrm{Pa}\cdot\mathrm{s}$ and $\omega_o = 3\,\mathrm{rad/s}$.

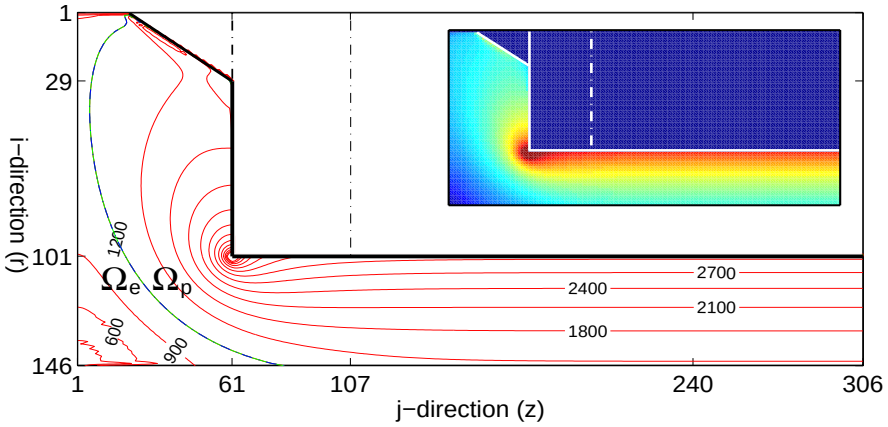


Figure 2.5: Profile of $\tau = \sqrt{(\mathbf{T} : \mathbf{T})/2}$ (in Pa) inside the ConTec **BML** Viscometer 3 when using the Bingham model $\eta = \mu + \tau_o/\dot{\gamma}$, with $\tau_o = 1200\,\mathrm{Pa}$, $\mu = 120\,\mathrm{Pa}\cdot\mathrm{s}$ and $\omega_o = 3\,\mathrm{rad/s}$.

the velocity profile of CPs with the CM coordinates at the corresponding spatial points $\mathbf{x} = (r, z)$. As shown with Equation 2.6, this type of velocity consists of the mass averaged velocity of all the solid particles composing the CP. Therefore, a spatial point in the figure $\mathbf{x} = (r, z)$ represents the average velocity of all the solid particles at and surrounding this point. As such, any random and discontinuous velocity contribution from the individual solid particles are summarized out, resulting in the smooth velocity profile shown in this figure. The above discussion applies equally for water. One cannot use the Navier-Stokes Equation B.18 (Appendix B.6) to predict the random (i.e. thermal) motion of the individual water molecule, because such a motion is on a scale below what can be provided by this equation. This lack of resolution (or sharpness) of the material space is a direct consequence of the steps done from Equation 2.2 to Equation 2.4. When going through these last-mentioned steps, another equation is also implicitly formed in the background. This equation deals only with the random and the spontaneous part of the velocity $\mathbf{v}_I$ of the individual

solid particles and is generally known as the internal energy equation (see Footnote 3, Page 378 and Footnote 6, Page 13). As shown with Equation B.41 (Page 391), this equation basically consist of $\rho\, c\, dT/dt = \eta\, \dot{\gamma}^2$.

2.4 Solid Particle Interaction and Viscosity

Background

It is clear from Section 2.2, that the CP consists of a large number of solid particles. These solid particles interact with each other by collisions that continuously occur between them (see Figure B.1, Page 378). For each collision that occurs, a momentum is transferred from one solid particle to the next. With this perception in mind, Bagnold [4] made a derivation of shear stress τ, where the momentum transfer from one solid particle to the next plays a primary role. In several papers [114, 113, 59, 93], his theory has been successfully used. In Bagnold's theory, only the effect from the linear momentum is taken into account. However, in some literature [114] ([113]), the effect of the angular momentum is also included. Sometimes, the angular momentum contribution is unimportant. For example, in the case of ideal gas, the shear stress can be calculated quite accurately without introducing such term into the mathematical analyze [93]. It is not clear how important the effect of angular momentum is to the cement paste. For simplicity reasons, its contribution is ignored here.

Objectives

In Section 2.4.1, a derivation of the shear stress τ and the shear viscosity η is made for a cement paste. This derivation is based on the above-mentioned momentum transfer. First, the direct momentum exchange between the larger cement particles is considered. This part is more or less a repetition from what Bagnold [4] made in his original paper. Thereafter, the effect of the smaller cement particles, which are in-between the larger ones, is considered. The main objective of the overall calculation presented here, is not to calculate the correct shear viscosity η, but rather to understand how it originates and how coagulation and hydration inflict its value (see discussion below, about coagulation and hydration). To keep the text here as simple as possible and still keeping the original objectives, some number of simplifications are made. Some of these are mentioned explicitly, while others are not.

Since the rheological properties of mortar and concrete are direct functions of the viscometric values of the cement paste (see for example [80]), the current text will also be valuable when considering the experimental results given in Chapter 6.

Coagulation (Reversible and Permanent)

The word "coagulation" will appear frequently and it describes the occurrence when two (or more) cement particles come into a solid-surface-to-solid-surface contact with each other for some duration of time; i.e. when the cement particles become "glued" to each other, and work is required to separate them. There are basically two kinds of coagulation. The first type is the **reversible coagulation**, where two coagulated cement particles can be separated (i.e. dispersed) again for the given rate of work $\dot{w}$ available to the suspension (see Equation B.34, Page 390). The second type of coagulation is the **permanent coagulation**, where the two cement particles cannot be separated for the given power $\dot{w}$ available. In either case of reversible coagulation

or of permanent coagulation, the effect on the shear viscosity η is the same, namely increasing its value.

In Section 2.4.2, a theory about the reversible coagulation is presented. Basically, this theory is about the bookkeeping of the number of reversible coagulated connections between the cement particles. These connections are often named "junctions" and is represented with the term J_{t}. Likewise, permanent connection is designated with $J_{\mathrm{t}}^{\mathrm{p}}$. Hence, the total number of connections between coagulated cement particles (in a suspension) is $J_{\mathrm{t}}^{\mathrm{tot}} = J_{\mathrm{t}}^{\mathrm{p}} + J_{\mathrm{t}}$ (see also Figure 2.11 about $J_{\mathrm{t}}^{\mathrm{p}}$ and J_{t}).

Hydration

When cement particles and water are intermixed, thus forming a cement particle suspension, complex chemical reactions[16] start to occur, generally designated as hydration. In chemical terms, hydration is a reaction of an anhydrous compound with water, yielding a new compound, a hydrate. During the first 15 minutes after the first contact with water (the pre-dormant period), the hydrate quickly covers the cement particle surface. Basically, this hydrate consists of so-called C-S-H gel (of type E) and of ettringite. The thickness of the hydrate increases with time, at different rate, during both the pre-dormant and the dormant[17] period. The thickness of the C-S-H gel has been reported to be about 50 nm on an alite[18] surface after 15 minutes from water addition [134]. Roughly at the same time, stubby rods of ettringite are present, typically 250 nm long and 100 nm thick [131]. They appear both on the cement particle surface, and at some distance away, in the solution water between the cement particles [131]. They are more abundant near the surface of the aluminate phase, and appear to nucleate in the solution and on the outer surface of the pre-mentioned layer of C-S-H gel [131]. Figure 2.17 (Page 38) shows an example of how the hydrate covers the unhydrated part of the cement particle, as a function of time.

A more detailed description about the chemical reactions between cement particles and water, is presented in Section 2.5.1.

2.4.1 Shear Viscosity η of Cement Particle Suspension

Momentum Transfer Across a Plane δA

Consider an imaginary plane of area δA, inside a cement particle suspension (i.e. inside a cement paste) as shown with illustration **A** in Figure 2.6. The idea is to calculate the rate of momentum transfer $\Delta\dot{\mathrm{p}}_{[\delta\mathrm{A}]}$ across this plane, since divided by the area δA, it gives the shear stress τ [93, 36]. More precisely, the shear stress can be calculated according to $\tau = T_{31} = \Delta\dot{\mathrm{p}}_{[\delta\mathrm{A}]}/\delta A$, where $\Delta\dot{\mathrm{p}}_{[\delta\mathrm{A}]}$ is a x_1-component of collision momentum, that is transferred across the plane δA in the x_3-direction.

Since it is hard to work directly with the reality shown with illustration **A** in Figure 2.6, some geometrical simplifications are necessary. These simplifications do

[16] Chemical shrinkage is a general phenomenon for any chemical reactions, where the products have as smaller volume than the reactants. For the hydration process involved here, such chemical shrinkage applies. This leads to an external contraction of the cement suspension, during the pre-dormant and dormant period (i.e. during the first hours after water addition). Depending on the cement type, the w/c-ratio, type of admixture and so forth, this shrinkage varies in magnitude from about zero to roughly 0.5% of total volume, during the first 100 minutes (see for example [64, 65]).

[17] As shown in Figure 2.15, the dormant period starts after the pre-dormant period, and usually lasts for 1 to 3 hours.

[18] The unhydrated cement particle consists of four mineral component, named alite, belite, aluminate and ferrite (see Page 34), which of the alit is the most abundant (50-70% by mass).

not interrupt the main objectives here, namely to understand how shear viscosity η originates and how coagulation and hydration inflict its value. Illustration **C**, demonstrates the geometrical simplification applied.

As is discussed in Section 2.5.1, the cement particles are poly-dispersed in size, with diameter ranging from $1\,\mu$m to $100\,\mu$m. Illustration **A** in Figure 2.6, demonstrates such polydispersity. The white and black spherical particles shown in illustration **C**, are ideal representation of the larger cement particles in the suspension, say above $40\,\mu$m in diameter. As is mentioned in the DLVO part of this chapter (see Page 42), because of their larger inertia, the larger cement particles interact more with each other by a hard sphere collision factor, rather than by coagulation/dispersion interactions. That is, the white and black particles seldom/newer coagulate. It is rather the smaller cement particles (say below $40\,\mu$m in diameter) that are effected by their mutual potential energy V_{T} and therefore interact with each other by coagulation and dispersion. To begin with, only the larger cement particles are treated. It will not be until on Page 25, that the effect of the smaller cement particles are considered.

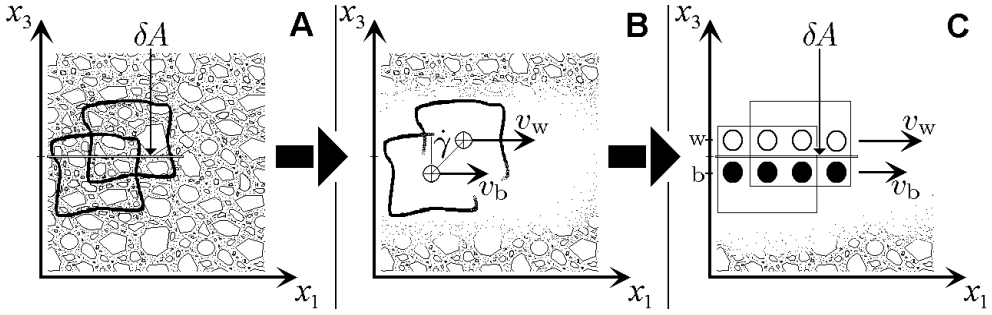


Figure 2.6: Two CPs and a plane δA (the horizontal line) inside a suspension. All three illustrations are of the same system: Illustration **A** represents a real condition of the suspension, while illustration **C** represents simplified and idealized picture of this reality. Illustration **B** demonstrates that the two velocities $v_{\mathrm{w}}\,\mathbf{i}_1$ and $v_{\mathrm{b}}\,\mathbf{i}_1$ are calculated according to Equation 2.6.

The Assumed Velocity Profile: $\mathbf{v} = v_1(x_3,t)\mathbf{i}_1$

It will be assumed that the flow takes place in layers that are in motion relative to one another, as a result of the shear rate $\dot{\gamma}$ shown with illustration **B** in Figure 2.6. In other words, the velocity profile of the suspension is assumed to be $\mathbf{v} = v_1(x_3,t)\mathbf{i}_1$. Hence, the strain rate tensor becomes $\dot{\boldsymbol{\varepsilon}} = [\partial v_1(x_3,t)/\partial x_3](\mathbf{i}_3\mathbf{i}_1 + \mathbf{i}_1\mathbf{i}_3)/2$ (see Equation 2.20) and accordingly, the shear rate develop into $\dot{\gamma} = |\partial v_1/\partial x_3| = \partial v_1/\partial x_3 \geq 0$ (see Equation 2.24).

Velocity of the Two Layers: $v_{\mathrm{w}}\,\mathbf{i}_1 \wedge v_{\mathrm{b}}\,\mathbf{i}_1$

The velocity of the white and black particles (the two layers of large cement particles under consideration) will be designated with v_{w} and v_{b}, respectively. These two velocity components are extracted from the real condition of illustration **A** in Figure 2.6. More precisely, the two velocity vectors $\mathbf{v}|_{(\mathrm{x_3=w})} = v_{\mathrm{w}}\,\mathbf{i}_1$ and $\mathbf{v}|_{(\mathrm{x_3=b})} = v_{\mathrm{b}}\,\mathbf{i}_1$ are the mass averaged velocity of two CPs, with a center of mass (CM) position at the same height x_3, as the white and the black particles, respectively. Hence, the velocity components v_{w} and v_{b} are calculated according to Equation 2.6. This is shown with illustration **B** in Figure 2.6.

Smooth and Random Velocity Contribution

As mentioned in Footnote 3 (Page 378), the velocity $\mathbf{v}_\mathrm{I}$ of a single solid particle, labeled with I, consist of two components, namely $\mathbf{v}$ and $\mathbf{v}_\mathrm{I} - \mathbf{v}$ (that is, $\mathbf{v}_\mathrm{I} = \mathbf{v} + [\mathbf{v}_\mathrm{I} - \mathbf{v}]$). The former velocity component is the smooth velocity of a CP at the same center of mass (CM) location as the solid particle in question, calculated according to Equation 2.6. The latter velocity component, namely $\mathbf{v}_\mathrm{I} - \mathbf{v}$, is the random velocity contribution of this same solid particle. Here, the random velocity contribution is mainly responsible for the (direct) collisions between the solid particles.

Focusing on Two Particles

Focusing now only on two large cement particles that are passing by each other, one white and the other black, as shown with illustration **A** in Figure 2.7. In addition to their colors, these two particles are labeled as I and I + 1, respectively. In the same fashion as mentioned above, each of them will have two velocity component. One is of the smooth type and the other of the random type. For the white particle, its velocity is $\mathbf{v}_\mathrm{I} = v_\mathrm{w}\,\mathbf{i}_1 + [\mathbf{v}_\mathrm{I} - v_\mathrm{w}\,\mathbf{i}_1]$ and for the black particle, its velocity is $\mathbf{v}_{\mathrm{I}+1} = v_\mathrm{b}\,\mathbf{i}_1 + [\mathbf{v}_{\mathrm{I}+1} - v_\mathrm{b}\,\mathbf{i}_1]$. Concentrating on the bypass of the white particle relative to the black particle, as shown with illustration **B** in Figure 2.7. In this frame of reference, the black particles is motionless while the white particles has the smooth velocity contribution $\mathbf{v}_\mathrm{smooth} = \mathbf{v}|_{(\mathrm{x}_3=\mathrm{w})} - \mathbf{v}|_{(\mathrm{x}_3=\mathrm{b})} = (v_\mathrm{w} - v_\mathrm{b})\,\mathbf{i}_1$ and the random velocity contribution $\mathbf{v}_\mathrm{rand} = (\mathbf{v}_\mathrm{I} - v_\mathrm{w}\,\mathbf{i}_1) - (\mathbf{v}_{\mathrm{I}+1} - v_\mathrm{b}\,\mathbf{i}_1)$. Hence, the motion of the white particle relative to the black one, is $\mathbf{v}_\mathrm{smooth} + \mathbf{v}_\mathrm{rand} = \mathbf{v}_\mathrm{I} - \mathbf{v}_{\mathrm{I}+1}$.

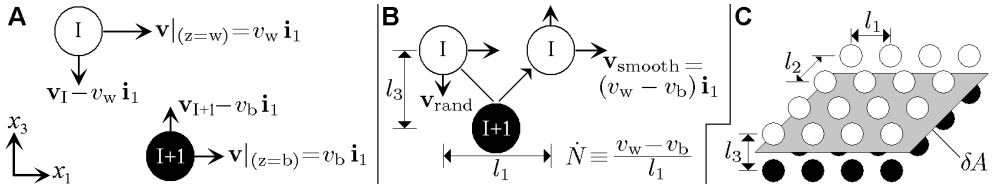


Figure 2.7: To the left: Each solid particle has two velocity component, one of the smooth type and the other of a random type. Center: Same as the left illustration, however motion is presented relative to the black particle. To the right: Rearrangement of the white and the black particles (i.e. the larger cement particles) in the vicinity to the plane δA. The lengths l_1, l_2 and l_3 are assumed to be larger than the diameter of the black and white particles.

Number of Bypass per Unit Time $\dot{N}$

In the above-mentioned frame of reference, the white particle will have the (smooth) bypassing speed of $v_\mathrm{w} - v_\mathrm{b}$ as shown with illustration **B** in Figure 2.7. As demonstrated with illustration **C** (and **B**) in the same figure, the average CM distance between the spherical particles is l_1 and l_3 in the x_1- and x_3-direction, respectively. With the relative bypassing speed of $v_\mathrm{w} - v_\mathrm{b}$ and the bypassing length of l_1, it is suitable to define a number of bypass per unit time: $\dot{N} \equiv (v_\mathrm{w} - v_\mathrm{b})/l_1$. For example, with $\dot{N} = 5\,\mathrm{s}^{-1}$ the white particle will travel the distance of $5\,l_1$, relative to the original black particle, during one second. That is, during the time interval of $1/\dot{N} = 0.2$ seconds, a single white particle will bypass a single black particle (i.e. travel the relative distance of l_1).

Number of Direct Collisions per Unit Bypass $\tilde{N}_{\mathrm{dc}}$

Although there is an average CM distance of l_1 and l_3 (and l_2 in the x_2-direction) between all the larger cement particles (see illustration **C** in Figure 2.7), each of them deviate constantly from this average distance, resulting in their collisions in-between. It is the random motion $\mathbf{v}_{\mathrm{rand}}$ that is mainly responsible for the collisions. Now, let $\tilde{N}_{\mathrm{dc}}$ be the average number of collisions that occur between a single white particle and a single black particle, as the former is bypassing the latter (i.e. as the former is traveling the distance of l_1, relative to the latter). That is, $\tilde{N}_{\mathrm{dc}}$ represents the number of direct collisions per unit bypass. The subscript "dc" is an acronym for "direct collision". With illustrations **A**, **B** and **C** in Figure 2.8, some examples of $\tilde{N}_{\mathrm{dc}}$ values are given.

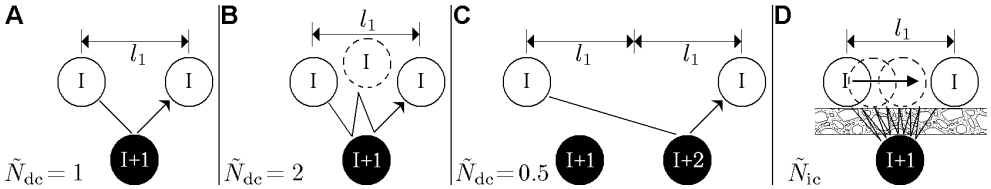


Figure 2.8: Illustrations **A**, **B** and **C** gives some examples of $\tilde{N}_{\mathrm{dc}}$ values. Illustration **D** demonstrates how the smaller solid particles, in-between the black and white particles, act as a conduit for an indirect and multiple collisions. The number of such indirect collisions per unit bypass, is designated here with $\tilde{N}_{\mathrm{ic}}$ (see discussion shortly).

Momentum Transfer by Direct Collisions $(\dot{N}\,\tilde{N}_{\mathrm{dc}})\,\Delta \mathrm{p}_{\mathrm{dc}}$

For a single collision between a white particle and a black particle (i.e. between a pair of large cement particles), the latter increases its momentum (in the average) by $\Delta \mathrm{p}_{\mathrm{dc}} = \alpha_{\mathrm{dc}}\, m\, (v_{\mathrm{w}} - v_{\mathrm{b}})$ in the x_1-direction. Similar contribution will also go in the x_2- and x_3-direction, depending on collision condition. However, for the sake of simplicity, the two last-mentioned contributions are ignored here. The interested reader is rather referred to the Bagnold's [4] original paper. With the momentum increase of $\Delta \mathrm{p}_{\mathrm{dc}} = \alpha_{\mathrm{dc}}\, m\, (v_{\mathrm{w}} - v_{\mathrm{b}})$ for the black particle, the white particle will lose this amount. It will be assumed that the white particles will quickly retain its original momentum after the impact, with the help of some external machinery. This is a common setup in viscometers. For example in the case of the ConTec **BML** Viscometer 3 (see Figure 3.1, Page 51), its outer cylinder is kept at rotational motion with the help of an external machinery. This results in that the outer most particles near the outer cylinder (or in this case, the white particles) do not lose momentum in the average, during the collision process with the inner ones (i.e. with the black particles).

The term m used in the momentum contribution $\Delta \mathrm{p}_{\mathrm{dc}} = \alpha_{\mathrm{dc}}\, m\, (v_{\mathrm{w}} - v_{\mathrm{b}})$, designates the (average) mass of the larger cement particles. The value α_{dc} is a material parameter, depending among other factors, on the cement particle surface roughness, and on angle of collision. See Bagnold's [4] original paper about the angle of collision. Increased hydration results in a larger surface roughness, which results in a better grip between two colliding particles. This gives a larger momentum transfer between two colliding particles and is mathematically described with a larger α_{dc} value.

With the number of bypass per unit time $\dot{N}$ and the number of direct collisions per unit bypass $\tilde{N}_{\mathrm{dc}}$, the product $\dot{N}\,\tilde{N}_{\mathrm{dc}} = \dot{n}_{\mathrm{dc}}$ describes the number of direct collisions per

unit time (or the direct collision rate), between a white particle and a black particle. Hence, the rate of momentum transfer due to direct collisions is $(\dot{N}\,\tilde{N}_{\mathrm{dc}})\,\Delta \mathrm{p}_{\mathrm{dc}}$.

Momentum Transfer by Indirect Collisions $(\dot{N}\,\tilde{N}_{\mathrm{ic}})\,\Delta \mathrm{p}_{\mathrm{ic}}$

Now, some consideration must be taken into account for the smaller cement particles, present in-between the white and black particles (i.e. between the larger cement particles). This condition is shown with illustration **D** of Figure 2.8. These smaller solid particles act as a conduit for "collisions" (or rather momentum transfer) that occur between a white and a black particle, without the latter actually bumping into each other. The number of such indirect collisions per unit bypass, is designated here with $\tilde{N}_{\mathrm{ic}}$. The subscript "ic" is an acronym for "indirect collision".

It is perhaps a bit difficult to describe in exact terms, what an indirect collision consist of. As a first approximation, one could describe such collision as a momentary formation of a single and continuous bridge of the smaller cement particles. Being simultaneously in contact with both a white and a black particle, such bridge would act as a conduit for momentum transfer between them. With larger degree of coagulation (right illustration of Figure 2.9), such type of bridges forms more frequently and are of a longer duration, relative to the seldom formed and shorter lived bridges of fully dispersed cement particles (center illustration of Figure 2.9). Hence, with increased coagulation, the value of $\tilde{N}_{\mathrm{ic}}$ will increase.

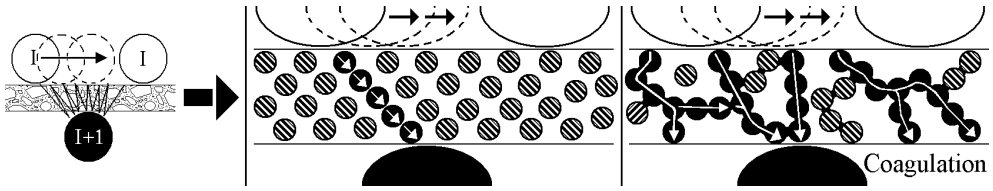


Figure 2.9: With larger degree of coagulation (right illustration), formation of bridges occurs more frequently resulting in a more frequent indirect collisions $\tilde{N}_{\mathrm{ic}}$ per unit bypass, than would have been a reality if these particles where fully dispersed (center illustration). Both a permanent ($J_{\mathrm{t}}^{\mathrm{p}}$) and a reversible ($J_{\mathrm{t}}$) junctions (or connections) can be present in the right illustration.

With the chemical reactions between the cement particles and water, the cement particles will grow in size, while the amount of free water[19] will be reduced. An example of such growth is shown in Figure 2.17 (Page 38). This will result in a reduced maneuverability of the individual smaller cement particle, in avoiding the formation of a continuous bridge between the white and black particles (see the center illustration of Figure 2.10). Hence, with increased hydration, the value of $\tilde{N}_{\mathrm{ic}}$ will increase. In addition to this, with large coagulation state $J_{\mathrm{t}}^{\mathrm{tot}} = J_{\mathrm{t}}^{\mathrm{p}} + J_{\mathrm{t}}$ (permanent and/or reversible), such bridges forms even more frequently and are of considerable longer duration. This is shown with the right illustration of Figure 2.10.

For each indirect collision that occur, the black particle will increase its momentum (in the average) by $\Delta \mathrm{p}_{\mathrm{ic}} = \alpha_{\mathrm{ic}}\, m\, (v_{\mathrm{w}} - v_{\mathrm{b}})$ in the x_1-direction, and correspondingly the

[19] Free water gives the possibility for the cement particles to have a certain average distance between them, as shown for example in Figure 2.9. As such, the free water is defined as the matrix. The free water (or rather the solution of free water and different ionic species) is not to be confused bound water (physically or chemically) to the surface of the cement particle. The total water inside the suspension is the sum of free water and physically bound water. In accordance with Reactions 2.47, 2.48, 2.49 and 2.50, the amount of total water is decreasing with increasing hydration; i.e. more and more water molecules of the total water, becomes chemically bound during the hydration process.

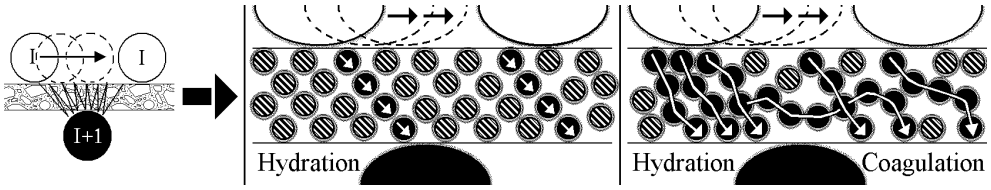


Figure 2.10: With larger degree of hydration (center illustration), formation of bridges occurs more frequently resulting in a more frequent indirect collisions $\tilde{N}_{\rm ic}$ per unit bypass, than would have been a reality if these particles where not hydrated at all (center illustration in Figure 2.9). In addition to hydration, with coagulated cement particles (right illustration), bridges will form even more frequently, resulting in a still larger $\tilde{N}_{\rm ic}$ value. Both a permanent ($J_{\rm t}^{\rm p}$) and a reversible ($J_{\rm t}$) junctions (or connections) can be present in the right illustration.

white particle will lose this amount. As before, the white particles will quickly retain its original momentum after the impact, with the help of some external machinery. The value $\alpha_{\rm ic}$ is a material parameter, depending among other factors, on the time duration of the pre-mentioned bridge and surface roughness of the cement particles. With increasing surface roughness, a better connection (or grip) exists between the larger cement particles and the bridges of smaller cement particles.

With the number of bypass per unit time $\dot{N}$ and the number of indirect collisions per unit bypass $\tilde{N}_{\rm ic}$, the product $\dot{N}\,\tilde{N}_{\rm ic} = \dot{n}_{\rm ic}$ describes the number of indirect collisions per unit time (or the indirect collision rate), between a white particle and a black particle. Hence, the rate of momentum transfer due to indirect collisions is $(\dot{N}\,\tilde{N}_{\rm ic})\,\Delta{\rm p}_{\rm ic}$.

Shear Stress τ and Shear Viscosity η

From the above, it is apparent that the rate of momentum transfer between a white and a black particle, becomes $\Delta\dot{\rm p} = (\tilde{N}_{\rm dc}\,\Delta{\rm p}_{\rm dc} + \tilde{N}_{\rm ic}\,\Delta{\rm p}_{\rm ic})\,\dot{N}$. Since the number of white-black-particle-pair is $\delta A/(l_1\,l_2)$ (see illustration **C** in Figure 2.7), the total rate of momentum transfer across the plane δA is $\Delta\dot{\rm p}_{[\delta{\rm A}]} = [\delta A/(l_1\,l_2)]\,\Delta\dot{\rm p}$. Hence, the shear stress is calculated as $\tau = \Delta\dot{\rm p}_{[\delta{\rm A}]}/\delta A = \Delta\dot{\rm p}/(l_1\,l_2)$. Assuming an equal CM distance between the white and black particles $l_1 = l_2 = l_3 = l$, the bypassing rate becomes $\dot{N} = (v_{\rm w} - v_{\rm b})/l_3 \approx \partial v_1/\partial x_3 = \dot{\gamma}$. Therefore, the shear stress becomes equal to $\tau = \Delta\dot{\rm p}/l^2 \approx [(\tilde{N}_{\rm dc}\,\Delta{\rm p}_{\rm dc} + \tilde{N}_{\rm ic}\,\Delta{\rm p}_{\rm ic})/l^2]\,\dot{\gamma}$. Now, using[20] $\tau = \eta\,\dot{\gamma}$, the shear viscosity can be extracted and is given by Equation 2.25.

$$\eta \approx (\tilde{N}_{\rm dc}\,\Delta{\rm p}_{\rm dc} + \tilde{N}_{\rm ic}\,\Delta{\rm p}_{\rm ic})/l^2 \tag{2.25}$$

In removing all the black and white particles, by taking the limit $l \to \infty$, the above shear viscosity becomes zero. This is an incorrect result because the remaining smaller cement particles, present in-between the previous black and white particles, will always make their own shear viscosity contribution. Again, in keeping the current text as simple as possible, their contribution (and others) will not be calculated here. In spite of its limitation, Equation 2.25 will suffice for the discussion that now follows.

Experimental Shear Viscosity η of Cement Particle Suspension

In Chapter 9, the (thixotropic) shear viscosity η of cement paste is extracted. This is done by a combination of computational calculations and experimental measure-

[20] See the paragraph just above Equation 7.5 on Page 156.

ments. The objectives of this particular chapter is to test the Hattori-Izumi theory (Section 2.4.2) and its modification (Section 9.3). In addition to this, the functional form of the shear viscosity function $\eta = \eta(\mathbf{x}, t)$ that is created, is of great interest. Finally, some supplementary information is extracted, about the effects of the polymers used in this thesis. In reproducing the behavior of the cement pastes by numerical means, the shear viscosity function had to have the form presented in Equation 2.26.

$$\eta = [\mu + \tilde{\mu} + m\,\dot{\gamma}^n] + \frac{[\tau_o + \tilde{\tau}_o]}{\dot{\gamma}} \tag{2.26}$$

With the idea behind Equation 2.25 in mind, some discussion about the viscometric parameters μ, τ_o, $\tilde{\mu}$ and $\tilde{\tau}_o$ can be made. This is done with the following items. In this explanation, illustrations **A**, **B**, **C** and **D** in Figure 2.11 is provided as an aid.

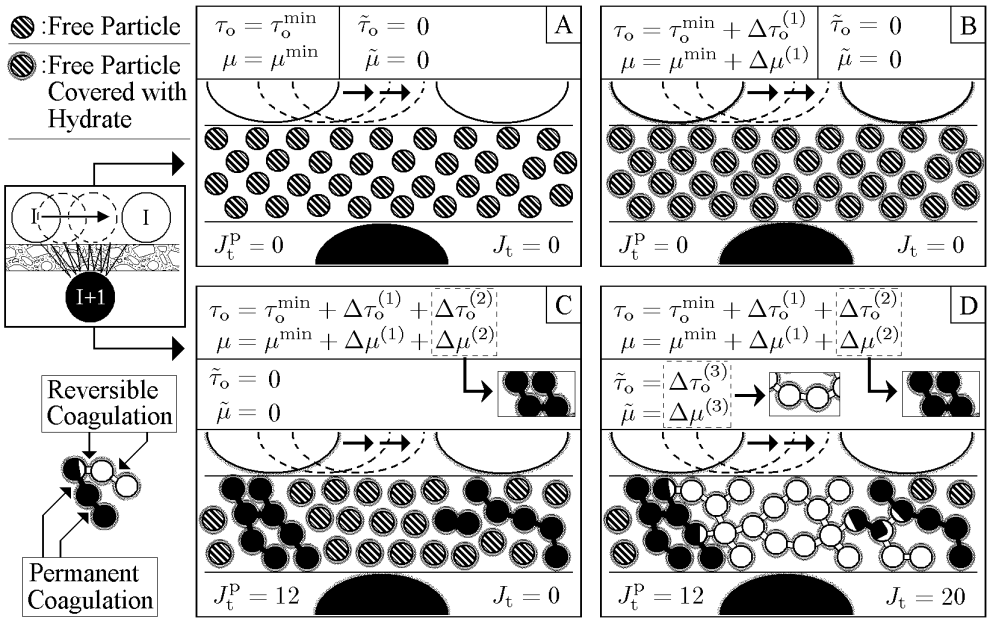


Figure 2.11: Explanation of the viscometric values μ, $\tilde{\mu}$, τ_o and $\tilde{\tau}_o$ in Equation 2.26. Note that the values μ^{min}, τ_o^{min}, $\Delta\mu^{(1)}$ and $\Delta\tau_o^{(1)}$ are related to the contribution of free particles. Since the number of such particles are reducing in illustrations **C** and **D**, then so are the values of these last-mentioned viscometric parameters.

- Illustration **A**: The parameters μ and τ_o are related to the basic momentum transfer between the white and black particles (i.e. the larger cement particles), when the smaller cement particles (in-between) are fully dispersed. These are the direct collisions that occur between the white and black particles (see also Figure 2.8) and the indirect collisions between them. As explained previously, the latter type of collision consist of momentum transfer through a short lived bridges that act as a momentum conduit (see center illustration of Figure 2.9).

- Illustration **B**: With the cement particle growth (due to hydration), the number of indirect collision per unit bypass $\tilde{N}_{ic}$ increases. This is because of how the bridges of the smaller cement particles form more easily as a result of the

concomitant loss in maneuverability in avoiding such formation (see center illustration of Figure 2.10). Also, increased hydration results in a larger surface roughness, which again results in a better grip between two colliding cement particles. This gives a larger momentum transfer Δp_{dc} and Δp_{ic} between two such colliding particles. All the above results in a larger τ_o and/or μ values.

- Illustration **C**: In addition to the above point, when the cement particles undergo a *permanent coagulation*, a larger occurence of the above-mentioned bridges is a reality. This results in a still larger τ_o and/or μ values. The size and number of the bridges are related to the permanent junction number J_t^p.

- Illustration **D**: The viscometric parameters $\tilde{\mu}$ and $\tilde{\tau}_o$ are related to the increased momentum transfer between the white and black particles, with bridges that can be broken down at later times. The size and number of such bridges are directly related to the reversible junction number J_t. Since the reversible junction number can increase and decease, then so can also the $\tilde{\mu}$ and $\tilde{\tau}_o$ values. Both illustrations **C** and **D** can correspond to the right illustration of Figure 2.10.

As Equation 2.26 is based on experimental results, the isolated shear viscosity contribution of the smaller cement particle, which was not included in Equation 2.25, is now taken into account. The term $m\,\dot{\gamma}^n$ in Equation 2.26, is sometimes included in the numerical simulations, to account for shear-thickening effect. This typically results from dilatancy behavior of the test material [9].

2.4.2 The Hattori-Izumi Theory

In this section, only the relevant part of the Hattori-Izumi theory [42, 41] is presented. For further information about this theory, the reader must consult with the original papers. The notation of the individual variables is kept as unchanged as possible from the authors. Here in Section 2.4.2, the word "coagulation" will appear frequently. With this it is meant the reversible coagulation only, where two coagulated cement particles can be separated again for the given rate of work $\dot{w}$ available to the suspension (see Equation B.34, Page 390). The number of junctions created by a reversible coagulation, is designated with J_t. As has been mentioned previously, when the two cement particles cannot be separated, the phrase "permanent coagulation" is used. The number of junctions created by a permanent coagulation, is designated with J_t^p. Permanent coagulation or permanent junction, is not an issue in Section 2.4.2.

Shear Viscosity

Basically, the theory of Hattori and Izumi consists of calculating the shear viscosity η for a cement particle suspension. It is assumed that the shear viscosity η is mostly dependent on the reversible coagulation state J_t. More precisely, the Hattori-Izumi theory assumes a shear viscosity of the form shown with Equation 2.27 [42, 41].

$$\eta_{HI} = B_3\, J_t^{2/3} + \{\text{other less relevant terms}\} \approx B_3\, J_t^{2/3} \tag{2.27}$$

The term B_3 is a so-called friction coefficient between the primary particles and has the physical unit of $[\mathrm{N}\cdot\mathrm{s}]$. The important variable J_t is explained below.

Junctions Between Primary Particles

The term $J_t \equiv J_3$, in Equation 2.27, represents the number of junctions (or connections) between the primary particles, per unit volume (see Figure 2.12). As such, this variable has the physical unit of $[m^{-3}]$ (as applies for J_t^{tot} and J_t^{p}). The number of primary particles per unit volume is represented with n_3. It is assumed that the value of J_t increases by a naturally occurring perikinetic[21] coagulation only. Also, it is assumed that the cement particles dissociate (or disperse) by orthokinetic[22] process. Consequently, reduction in junctions J_t results by a stirring process only.

As shown with Figure 2.12, two initial states of suspension can be defined. One consists of a completely dispersed state (illustration **A**) and the other of a completely coagulated state (illustration **C**). It is assumed that the coagulated particles form a kind of open-chain-structure, rather than closely interconnecting spherical/cubical structure (with multiple internal cross-junctions). If coagulation occurs fast, then this could be an acceptable assumption, since rapid coagulation (i.e. a large H) forms a very loose and open structure, with a large amount of entrained water [52]. Slowly coagulated systems may take much longer time to form, but it will be much more compact [52].

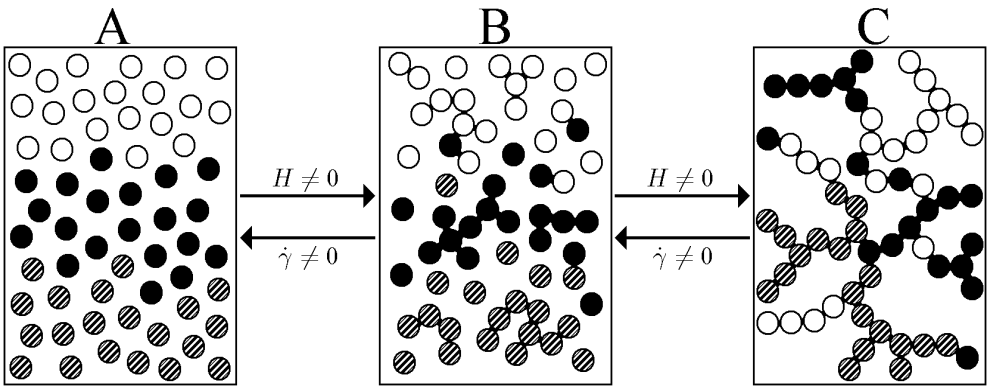


Figure 2.12: Relationship between number of primary particles n_3, number of particles n_t and junctions J_t (assuming a reference volume of $1\,m^{-3}$): **A)** $n_t = 60\,m^{-3}$, $J_t = 0\,m^{-3}$, $n_3 = n_t + J_t = 60\,m^{-3}$ and $U_3 = J_t/n_3 = 0$. **B)** $n_t = 27\,m^{-3}$, $J_t = 33\,m^{-3}$, $n_3 = n_t + J_t = 60\,m^{-3}$ and $U_3 = J_t/n_3 = 0.55$. **C)** $n_t = 1\,m^{-3}$, $J_t = 59\,m^{-3}$, $n_3 = n_t + J_t = 60\,m^{-3}$ and $U_3 = J_t/n_3 = 0.98 \approx 1$. Illustration **A** corresponds to illustrations **A**, **B** and **C** in Figure 2.11. Illustration **C** corresponds to illustration **D** in Figure 2.11.

As is demonstrated with Figure 2.12, the relationship between number of primary particles n_3, number of particles n_t and junctions J_t is given by the following:

$$n_3 = n_t + J_t \tag{2.28}$$

All the above variables have the physical unit of $[m^{-3}]$. This means that the term n_3 represents the number of **primary** particles per unit volume, and n_t the number of particles per unit volume.

Here, the perikinetic coagulation rate theory, established by Verwey and Overbeek is employed [141, 42, 41]. In their calculation, the decreasing number of particles n_t

[21] **Perikinetic** ⇒ no stirring ⇒ $\dot{\gamma} = 0$.
[22] **Orthokinetic** ⇒ with stirring ⇒ $\dot{\gamma} \neq 0$.

(in the suspension) is expressed as follows:

$$-\frac{dn_\mathrm{t}}{dt} = \frac{H\, n_\mathrm{t}^2}{n_3} \qquad (\dot{\gamma} = 0) \tag{2.29}$$

The constant H in the above, is known as the coagulation rate constant. Integrating this equation from $n_\mathrm{t}(0) = n_3$ to $n_\mathrm{t}(t) = n_\mathrm{t}$, results in the equation shown below.

$$n_\mathrm{t} = \frac{n_3}{H\, t + 1} \qquad (\dot{\gamma} = 0) \tag{2.30}$$

Combining Equation 2.28 and 2.30, yields the important result:

$$J_\mathrm{t} = n_3 \left(\frac{H\, t}{H\, t + 1} \right) \qquad (\dot{\gamma} = 0) \tag{2.31}$$

Because of the use of Equation 2.30, the above equation applies only when the primary particles n_3 are completely dispersed at $t = 0$. That is, at $t = 0$ there exists no junctions ($J_\mathrm{t}|_{t=0} = 0$) and hence all the particles n_t consist of primary particles ($n_\mathrm{t} = n_3$). This situation is presented with illustration **A** in Figure 2.12.

For a completely coagulated system ($n_\mathrm{t}|_{t=0} = 1$), the authors argue [42, 41] that the reduction in junctions J_t due to stirring ($\dot{\gamma} \neq 0$) must result in a kind of opposite functional relationship to what is shown with Equation 2.31. As such they come up with the conclusion that the reduction in junctions J_t is given by the following equation [42, 41]:

$$J_\mathrm{t} = n_3 \left(1 - \frac{\dot{\gamma}\, t}{\dot{\gamma}\, t + 1} \right) \qquad (H = 0) \tag{2.32}$$

The above applies only when the primary particles n_3 are completely coagulated at $t = 0$. That is, at $t = 0$ there exists no isolated primary particle ($n_\mathrm{t} = 1$) and hence the number of junctions is equal to the number of primary particles ($J_\mathrm{t}|_{t=0} = n_3 - n_\mathrm{t} = n_3$, c.f. $n_3 \gg n_\mathrm{t} = 1$). This situation is presented with illustration **C** in Figure 2.12.

Equations 2.31 and 2.32 apply to totally different initial states: The former equation assumes complete dispersion at $t = 0$, while the latter assumes a complete coagulation. Also, they apply to totally different physical conditions: The former equation applies when the condition $H \neq 0 \wedge \dot{\gamma} = 0$ is valid, while the latter applies when the opposite condition $H = 0 \wedge \dot{\gamma} \neq 0$ is valid. However, in spite of their differences, by using those two equations in a certain combination the authors conclude, with a small paragraph, that the total number of junction J_t for an intermediate state, must comply with Equation 2.33.

$$J_\mathrm{t} = \frac{n_3\, [U_\mathrm{o}(\dot{\gamma}\, H\, t^2 + 1) + H\, t]}{(H\, t + 1)(\dot{\gamma}\, t + 1)} \tag{2.33}$$

The term $J_\mathrm{o} = n_3\, U_\mathrm{o}$ in the above, is the junction number that exists at the start of observation ($t = 0$), which is the condition shown with illustration **B** in Figure 2.12. For the above equation, coagulation and dispersion can now be occurring at the same time ($H \neq 0 \wedge \dot{\gamma} \neq 0$).

What now follows is a more detailed and comprehensive derivation of Equation 2.33, than is made in the original paper. After a number of mathematical steps, the same result is nevertheless produced. As was done in the original paper, the number of junctions originating from the n_o particles, namely J_1, is first calculated. Thereafter, the junctions originating from the J_o junctions, namely J_2, is found. In the end, the total number of junctions $J_\mathrm{t} = J_1 + J_2$ is calculated.

The Junction Number J_1 (Junctions Originating From the n_o Particles)

As mentioned previously, Equation 2.32 applies when the primary particles n_3 are completely coagulated at $t = 0$; i.e. at $t = 0$, the number of junctions J_t is equal to the number of primary particles $J_t|_{t=0} = n_3 - n_t = n_3$ (c.f. $n_3 \gg n_t = 1$). Using this fact in this last-mentioned equation, gives Equation 2.34.

$$J_t = J_t|_{t=0}\left(1 - \frac{\dot{\gamma}\, t}{\dot{\gamma}\, t + 1}\right) \qquad (H = 0) \tag{2.34}$$

For the given physical condition $H \neq 0$ and $\dot{\gamma} \neq 0$, a suspension usually never consists of complete coagulated or of complete dispersed state. Rather it is in an intermediate state, where the number of junctions is J_o and the number of particles is n_o (see illustration **B** in Figure 2.12). With this, it is convenient to redefine the time axis in such a way that $J_t = J_o$ and $n_t = n_o$ at the time $t_e = 0$ (that is, $J_t|_{t_e=0} \equiv J_o$; $n_t|_{t_e=0} \equiv n_o$; $U_3|_{t_e=0} \equiv U_o$). This time axis is called the "experimental time" in [42, 41] and hence the subscript "e" in the time notation t. With the above, the same relationship as is presented in Figure 2.12 can be produced (see also Equation 2.28):

$$J_t + n_t = n_3 \quad \wedge \quad U_3 = J_t/n_3 \quad \Rightarrow \quad J_o + n_o = n_3 \quad \wedge \quad U_o = J_o/n_3 \tag{2.35}$$

One can make direct use of Equation 2.31 for this intermediate state, to calculate the additional junctions J_1^{new} (to the already existing junctions J_o), created due to coagulation ($H \neq 0$). This process is shown with illustration **A** and **B** in Figure 2.13.

$$J_1^{new} = n_o\left(\frac{H\, t_e}{H\, t_e + 1}\right) \qquad \Rightarrow \text{Condition } \mathbf{A} \rightarrow \mathbf{B} \text{ in Figure 2.13} \tag{2.36}$$

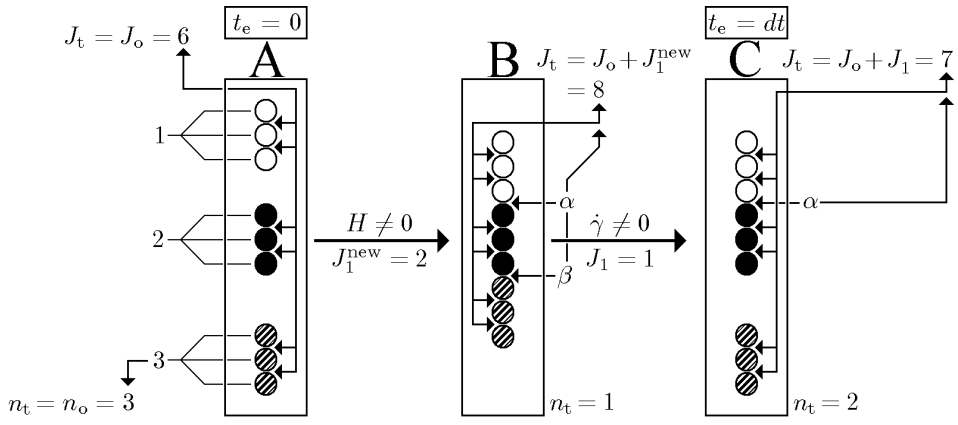


Figure 2.13: $\mathbf{A} \rightarrow \mathbf{B}$: Because of the coagulation $H \neq 0$, additional two new junctions are created $J_1^{new} = 2$, namely the junctions α and β. Then there afterwards, $\mathbf{B} \rightarrow \mathbf{C}$, the junction β is terminated because of the agitation $\dot{\gamma} \neq 0$. Note that the two steps $\mathbf{A} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{C}$ are mathematical ones. The physical step consists of $\mathbf{A} \rightarrow \mathbf{C}$ (from $t_e = 0$ to $t_e = dt$).

As applies for Equation 2.31, Equation 2.36 is only valid when the condition $\dot{\gamma} = 0$ applies. Hence, it makes an overestimation of the number of junctions created. Because of the concomitant agitation ($\dot{\gamma} \neq 0$), some of the calculated junctions J_1^{new}

are never actually formed. However, by using J_1^{new} as an initial condition in Equation 2.34, the effect of agitation (or stirring) is taken into account for:

$$J_1 = J_1^{\text{new}} \left(1 - \frac{\dot{\gamma}\, t_e}{\dot{\gamma}\, t_e + 1}\right) \qquad \Rightarrow \text{Condition } \mathbf{B} \rightarrow \mathbf{C} \text{ in Figure 2.13} \qquad (2.37)$$

Now, putting the outcome from Equation 2.36 into the above, results in Equation 2.38.

$$J_1 = n_o \left(\frac{H\, t_e}{H\, t_e + 1}\right)\left(1 - \frac{\dot{\gamma}\, t_e}{\dot{\gamma}\, t_e + 1}\right) = \frac{n_o\, H\, t_e}{(H\, t_e + 1)(\dot{\gamma}\, t_e + 1)} =$$
$$= \frac{n_3(1 - U_o) H\, t_e}{(H\, t_e + 1)(\dot{\gamma}\, t_e + 1)} \qquad \Rightarrow \text{Condition } \mathbf{A} \rightarrow \mathbf{C} \text{ in Figure 2.13} \qquad (2.38)$$

The last part in the above equation is gained by using the relationship $n_o = n_3 - J_o = n_3(1 - U_o)$ as described by Equation 2.35. Note that the two separated steps in Equations 2.36 and 2.37 ($\mathbf{A} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{C}$) are mathematical ones. This is because of their difference in physical condition, to which they apply. The physical step consists of $\mathbf{A} \rightarrow \mathbf{C}$ and occurs over the time period from $t_e = 0$ to $t_e = dt$.

The Junction Number J_2 (Junctions Originating From the J_o Junctions)

From Equation 2.38, one could suggest that the total number of junctions J_t for a suspension, is given by $J_t = J_1 + J_o$. This equation would only be valid if the n_o particles could never disperse (i.e. disconnect) to reduce the total number of junctions J_t. Such a case is shown with Figure 2.13. However, Figure 2.14 demonstrates how the original n_o particles can (internally) disperse, to increase their number from $n_t = n_o = 3$ to $n_t = n_o + n_{\text{add}} = 6$ ($\mathbf{A} \rightarrow \mathbf{B}$), and then to $n_t = 4$ ($\mathbf{B} \rightarrow \mathbf{C}$).

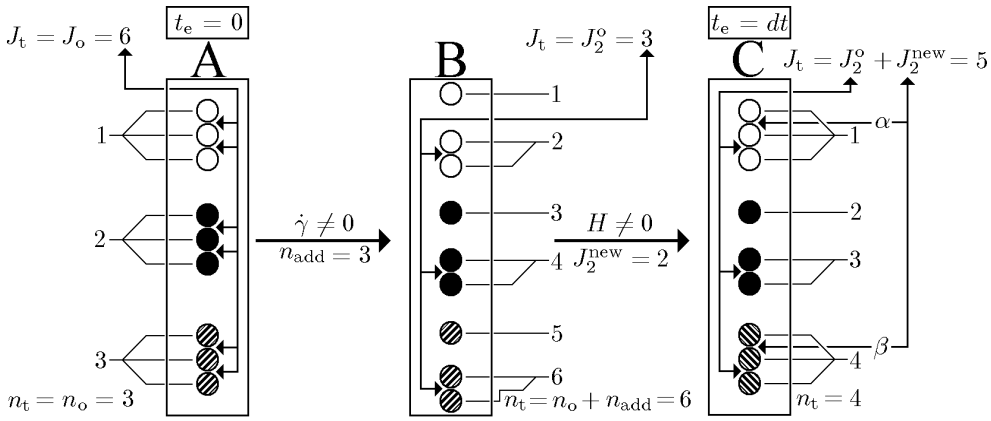


Figure 2.14: $\mathbf{A} \rightarrow \mathbf{B}$: Because of agitation $\dot{\gamma} \neq 0$, the original n_o particles disperse (or disconnect), to increase their number from $n_t = n_o = 3$ to $n_t = n_o + n_{\text{add}} = 6$ ($n_{\text{add}} = 3$). Then there afterwards, $\mathbf{B} \rightarrow \mathbf{C}$, the total number of particles are reduced because of the coagulation $H \neq 0$. Note, as before, the two steps $\mathbf{A} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{C}$ are mathematical ones. The physical step consist of $\mathbf{A} \rightarrow \mathbf{C}$ (from $t_e = 0$ to $t_e = dt$).

When the original n_o particles disperse (internally) to increase their number from $n_t = n_o$ to $n_t = n_o + n_{\text{add}}$, a reduction in junctions must occur at the same time (condition $\mathbf{A} \rightarrow \mathbf{B}$ in Figure 2.14). As always stated, at the time $t_e = 0$, the total number of junctions is $J_t = J_o$ (illustration **A** in Figures 2.13 and 2.14). Hence, the

reduction in junction from the original J_o junctions, due to (internal) dispersion of the original n_o particles, can be extracted from Equation 2.34:

$$J_2^o = J_o \left(1 - \frac{\dot{\gamma}\, t_e}{\dot{\gamma}\, t_e + 1}\right) \qquad \Rightarrow \text{Condition } \mathbf{A} \rightarrow \mathbf{B} \text{ in Figure 2.14} \tag{2.39}$$

Now, according to Equation 2.35, the following applies (see Figure 2.14):

$$J_o = n_3 - n_o \tag{2.40}$$

$$J_2^o = n_3 - (n_o + n_{add}) \tag{2.41}$$

The term n_{add} is the number of new particles, "created" as a consequence of the reduction in junctions from the original J_o junctions to the J_2^o junctions. Subtracting Equation 2.41 from Equation 2.40, with the concomitant use of Equation 2.39, results in a formula giving the number of these new particles:

$$n_{add} = J_o \left[1 - \left(1 - \frac{\dot{\gamma}\, t_e}{\dot{\gamma}\, t_e + 1}\right)\right] = J_o \left(1 - \frac{1}{\dot{\gamma}\, t_e + 1}\right) \tag{2.42}$$

Because of the condition $H \neq 0$, these new additional particles n_{add} will coagulate, resulting in a new additional junctions J_2^{new} (to the already existing junctions J_2^o). [Note that the ("external") coagulation or dispersion of the original n_o particles is not of concern here, because that case is already taken care of with J_1, as shown in Figure 2.13]. Using n_{add} as an initial condition in Equation 2.31, the new additional junctions J_2^{new} can be calculated as shown with Equation 2.43 (condition $\mathbf{B} \rightarrow \mathbf{C}$ in Figure 2.14).

$$J_2^{new} = n_{add} \left(\frac{H\, t_e}{H\, t_e + 1}\right) = J_o \left(1 - \frac{1}{\dot{\gamma}\, t_e + 1}\right) \left(\frac{H\, t_e}{H\, t_e + 1}\right) \tag{2.43}$$

Therefore, the change in junction number relative to the original J_o junctions is given as (condition $\mathbf{A} \rightarrow \mathbf{C}$ in Figure 2.14):

$$\begin{aligned} J_2 = J_2^o + J_2^{new} &= J_o \left(1 - \frac{\dot{\gamma}\, t_e}{\dot{\gamma}\, t_e + 1}\right) + J_o \left(1 - \frac{1}{\dot{\gamma}\, t_e + 1}\right) \left(\frac{H\, t_e}{H\, t_e + 1}\right) = \\ &= J_o \left[\frac{H\, \dot{\gamma}\, t_e^2 + H\, t_e + 1}{(H\, t_e + 1)(\dot{\gamma}\, t_e + 1)}\right] = n_3\, U_o \left[\frac{H\, \dot{\gamma}\, t_e^2 + H\, t_e + 1}{(H\, t_e + 1)(\dot{\gamma}\, t_e + 1)}\right] \end{aligned} \tag{2.44}$$

The last part in the above is gained by using the relationship $J_o = n_3\, U_o$ as is described by Equation 2.35.

The Final Junction Number $J_t = J_1 + J_2$

The processes presented with Figure 2.13 ($\Rightarrow J_1$) and Figure 2.14 ($\Rightarrow J_2$), occur simultaneously over the time period from $t_e = 0$ to $t_e = dt$. As such, the total number of junctions J_t is gained simply by adding[23] J_1 and J_2. Hence, from Equations 2.44 and 2.38, the total number of junctions J_t for a particle suspension, is given by:

$$J_t = J_1 + J_2 = \frac{n_3\, [U_o(\dot{\gamma}\, H\, t^2 + 1) + H\, t]}{(H\, t + 1)(\dot{\gamma}\, t + 1)} \tag{2.45}$$

[23] It should be clear that the contribution of J_o is included in J_2 term. This is apparent by putting $t_e = 0$ in Equation 2.44, giving $J_2 = J_o$. Therefore, the total junction number is not calculated as $J_t = J_1 + J_2 + J_o$, but rather as $J_t = J_1 + J_2$.

In the above, the experimental time t_e is replaced with t. Putting the above into Equation 2.27, gives the shear viscosity, predicted by Hattori and Izumi:

$$\eta_{\mathrm{HI}} = B_3 \left[\frac{n_3 \left[U_o(\dot{\gamma}\, H\, t^2 + 1) + H\, t \right]}{(H\, t + 1)(\dot{\gamma}\, t + 1)} \right]^{2/3} \tag{2.46}$$

Using this equation in a numerical simulation, to reproduce experimental observations, resulted in a complete failure (see Figure 9.1, Page 208). But as is demonstrated in Chapter 9, with some modifications, the above equation becomes a quite useful tool.

2.5 Coagulation Rate *H*

In this section, an overview of the relevant theories that are related to the coagulation rate H are presented. These include the chemical reactions between the cement particles and water (Section 2.5.1) and the concomitant change in potential energy V_T between the cement particles (Section 2.5.2) as a result of these reactions. Finally (Section 2.5.3), a relationship between the coagulation rate H and the potential energy V_T is presented for the perikinetic case ($\dot{\gamma} = 0$). In the end, some necessary modifications to the coagulation rate H are introduced, to apply for the orthokinetic case ($\dot{\gamma} \neq 0$). These modifications and the theory presented in Section 2.4.2, will be used in Chapter 9.

2.5.1 Chemical Reactions of the Cement Particle and Water

When cement particles and water are intermixed, thus forming a cement particle suspension (i.e. forming a cement paste), complex chemical reactions start to occur, generally designated as hydration. In strictly chemical terms, hydration is a reaction of an anhydrous compound with water, yielding a new compound, a hydrate [91]. As Portland cement is a multi-component system, its hydration is a rather complex process, consisting of a series of individual chemical reactions that take place both in parallel and successively [91]. For the sake of clarity, only the most important reactions are considered here. These are the ones related to coagulation rate H in the pre-dormant and dormant period. Also, the effect of lignosulfonate on the cement hydration is not discussed in this section. This is rather done in Section 6.4.1.

A further discussion about the overall complex chemical reactions involved in cement hydration, can for example be found in the textbook edited by Hewlett [91].

The Cement Clinker

Clinker consists of the combined minerals, which the cement particle is made from. It consists mainly of crystallized calcium silicates[24], C_3S and (β-) C_2S surrounded by interstitial calcium aluminate C_3A and calcium aluminoferrite C_4AF. Because of the impurities in these compounds, one often refers to their impure state: alite, belite, aluminate and ferrite, respectively [131]. The average magnitude of each constituents inside the cement particles is a variable depending on the cement type. Typically, the alite constitutes for about 50-70% by mass, while belite, aluminate and ferrite

[24] $C_3S \Rightarrow$ Tricalcium Silicate; $C_2S \Rightarrow$ Dicalcium Silicate; $C_3A \Rightarrow$ Tricalcium Aluminate; $C_4AF \Rightarrow$ Tetracalcium Aluminoferrite (a mean composition of a solid solution series ranging from C_6A_2F to C_6AF_2) [152]. Following abbreviations for the oxides are used: C = CaO; S = SiO_2; A = Al_2O_3; F = Fe_2O_3; H = H_2O; $\bar{S}$ = SO_3; N = Na_2O; K = K_2O; M = MgO.

constitutes only for about 15-30%, 5-10% and 5-15%, respectively [131]. The left illustration of Figure 2.15 shows a typical cement particle submerged in water [112]. In this figure the alite is the predominant clinker phase. The belite forms well defined regions that are rounded, striated and darker than the alite. The interstitial material present, for example in a vertical band left of center within the large particle, consist mainly of ferrite (light) and aluminate (dark). From this illustration, it is clear that the cement grain is not a perfect sphere as shown in Figure 2.12. However, it should be noted that no specific geometrical form is explicitly assumed in the theory presented in Section 2.4.2.

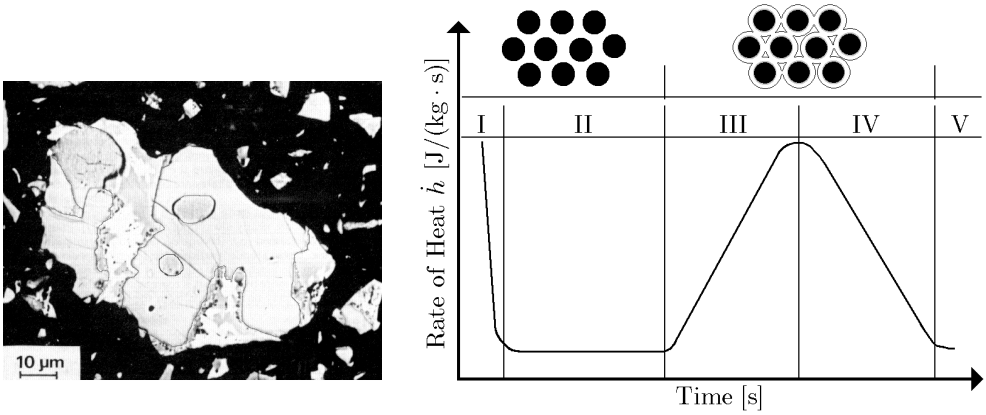


Figure 2.15: To the left: A typical cement grain in fresh paste [112]. To the right: Schematic figure displaying the heat of hydration $\dot{h}$ when cement particles and water are intermixed [152, 98, 29]. In Stage I, there is an initial rapid reaction, which is completed within about 15 minutes. It is followed by the dormant period, which lasts typically for about 1 to 3 hours (Stage II). First and final setting is in Stage III (typically occurring after 2 to 8 hours) and the early development of strength starts in the beginning at Stage IV. The rate of heat $\dot{h}$ is defined in Appendix B.7.2.

The creation of cement clinker consists of burning calcareous material, such as limestone $CaCO_3$, and of silica and alumina rich materials such as clay or shale [89]. First, the raw materials are crushed into very fine powder and then intimately mixed in predetermined proportions [68]. Thereafter the powder is burned in a large rotary kiln at a temperature of about 1400 to 1450°C where the material sinters and partially fuses into clinker [89, 131]. After rapid cooling, the clinker is ground in mills and intermixed with gypsum $C\bar{S}H_2$ (2.5-5% by total weight of cement) to control the hydration of C_3A. At first contact with water, the gypsum dissolves into Ca^{+2}, SO_4^{-2} and water. It is the sulfate ion SO_4^{-2} that controls the hydration of the C_3A compound [152].

Additional oxides are also present in the cement particle like K_2O, Na_2O ("the alkalis") and MgO, but only in minor amount. The alkalis are often found in the mineral $K_3N\bar{S}_4$ or as solid solution in the main minerals [61]. Free CaO (i.e. free lime) is also found in the cement particle due to insufficient reactions in the kiln, and/or due to the decomposition $C_3S \rightarrow C_2S + CaO$ [109, 110]. The latter reaction appears if the cooling rate of the clinker is too slow inside the kiln [110, 61].

The cement particles are poly-dispersed in size, with diameters ranging from 1 μm to 100 μm [131]. Typically 7-9% by weight are smaller than 2 μm and 0-4% by weight coarser than 90 μm [131]. The mass averaged size of the cement particles is reported to be typically 22 μm [118].

Hydration of the Cement Particle

Since all relevant reactions of Portland cement are exothermic, cement hydration is well described with the rate of heat development from isothermal calorimetry as a function of time [152, 98, 29]. A schematic illustration of such curve is shown in Figure 2.15. On first contact with water, a heat of hydration peak appears instantaneously as shown in this figure (Stage I). This peak is caused by dissolution of alkali sulfates $K_3N\bar{S}_4$, present in the clinker, which dissolves completely within seconds contributing K^+, Na^+ and SO_4^{-2} to the solution between the cement particles [91]. This is followed by the dissolution of Ca^{+2}, OH^- and SiO_4^{-4} ions from the C_3S and C_2S part of the cement grain [91, 152]. The pH-value rises to over 12 within few minutes [152]. This hydrolysis slows down quickly but still continues to some extent, throughout the dormant period (Stage II in Figure 2.15) [152]. As shown in Figure 2.16, a rapid change in concentration of the above ions is a reality in the early stages of hydration [132, 128].

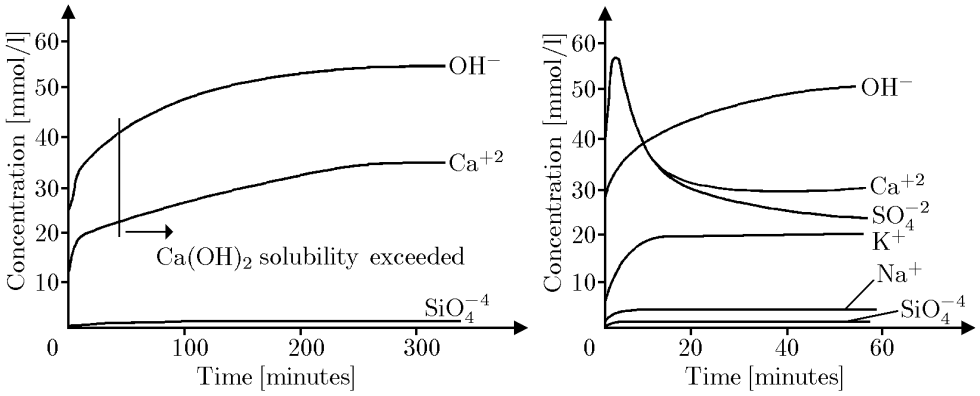


Figure 2.16: Concentration of ions in solution, plotted against time. To the left: Hydration of pure C_3S at $w/c = 5$. To the right: Hydration of ordinary Portland cement at $w/c = 2$. Both illustration are reproduced from the textbook of Tattersall and Banfill [128], however the findings are originally reported by Thomas et al. [132].

The first peak in Figure 2.15 (Stage I), consists also of ettringite formation (that is, Al_2O_3-Fe_2O_3-trisulfate, or simply AFt) according to Reactions 2.47 and 2.48 [152].

$$C_3A + 3C\bar{S}H_2 + 26H \rightarrow C_3A \cdot 3C\bar{S} \cdot H_{32} \tag{2.47}$$

$$3C_4AF + 12C\bar{S}H_2 + 110H \rightarrow 4[C_3(A,F) \cdot 3C\bar{S} \cdot H_{32}] + 2(A,F)H_3 \tag{2.48}$$

As shown with the two above reactions, C_3A and C_4AF are in competition for the available sulfate ions SO_4^{-2} present in the solution. Because the latter compound reacts much slower than the former, most of the sulfate will go in production of $C_3A \cdot 3C\bar{S} \cdot H_{32}$ by Reaction 2.47 [152]. This product is in most part created in the bulk solution, and quickly precipitates at the C_3A and C_4AF surface of the cement particle [29, 62]. The parentheses in Reaction 2.48 stand for when Al can partly or completely be replaced by Fe. Both above reactions are slowed down as AFt phases create a diffusion barrier between the water and the C_3A and C_4AF part of the cement grain and stays at a minimum through the dormant period (Stage II) [152].

Stage III in Figure 2.15, consists of stiffening that occurs within the cement paste after few hours, followed by Stage IV where the subsequent development of strength starts [152]. Both phenomena are due to the following reactions [152]:

$$2C_3S + 7H \rightarrow C_3S_2H_4 + 3CH \tag{2.49}$$

$$2C_2S + 5H \rightarrow C_3S_2H_4 + CH \tag{2.50}$$

The stoichiometric Reactions 2.47 to 2.50 are ideal, meaning that they resemble an average reactions. For example, in Reactions 2.49 and 2.50, the stoichiometric values 3, 2 and 4 in the product $C_3S_2H_4$ can vary. A more accurate presentation of the product would be $C_aS_bH_c$ where the values a, b and c generally vary considerably with w/c-ratio and time after water addition [63]. It has been shown that the product C-S-H is in the form of extremely small interlocking crystals that are measured in nanometers [14]. Therefore the C-S-H structure is often described as gel [88]. As stated above, the Reactions 2.49 and 2.50 start to occur in their full rate, in the end of dormant period. However, during the pre-dormant period, the dissolution rate of the C_3S compound is faster than diffusion can carry the dissolved ions (Ca^{+2}, OH^-, SiO_4^{-4}) away from the C_3S surface, causing a concentration gradient in its vicinity [91]. Here, the liquid phase near the surface, quickly becomes oversaturated with respect to calcium silicate hydrate and a layer of a C-S-H product (of type E [35]) start to precipitate at this surface [91, 128, 35]. In this last-mentioned theory, it is assumed that all chemical species of the C_3S mineral, which are in direct contact with water, dissolve into the solution. This process is known as congruent dissolution. However, there is another theory present to this, which does not involve congruent dissolution [91, 35]: Initial dissolution is rather considered to be incongruent, with Ca^{+2} and OH^- moving rabidly into the solution and leaving a silica-rich surface layer. Subsequent re-adsorption of Ca^{+2} on the now negatively charged surface creates an electrical double-layer [91, 35]. Continued incongruent dissolution should increase the thickness of this silica-rich layer, which eventually reorganizes into C-S-H [35].

Since the overall chemistry of the C_2S is similar to the C_3S, however at a slower rate [91, 152], similar chemical occurrence could be expected at the C_2S surface.

Summary

From the above text, it appears that in the pre-dormant period, the cement particles are coated by hydration products, namely of both AFt and C-S-H products. The C-S-H product is of type E and continues to develop on the C_3S and C_2S surface, through the dormant period [35, 91, 152]. The formation of the AFt and C-S-H products are slowed down trough the dormant period [152, 91].

Trettin [134] used an atomic force microscopy (AFM) to show that the (isolated) alite crystals build up about 50 nm thick C-S-H layer, during the pre-dormant period. During the same time period, stubby rods of ettringite (i.e. AFt) develops, typically 250 nm long and 100 nm thick [131]. They appear both on the cement particle surface, and at some distance away, in the solution water between the cement particles [131]. They are more abundant near the surface of the aluminate phase, and appear to nucleate in the solution and on the outer surface of the above-mentioned layer of C-S-H product [131]. Figure 2.17 [111] shows an example of how the hydrate covers the unhydrated part of the cement particle, as a function of time.

The most significant events occurring in the pre-dormant and dormant period, may be viewed as the hydrate coating of the cement particles and the solubilization of a

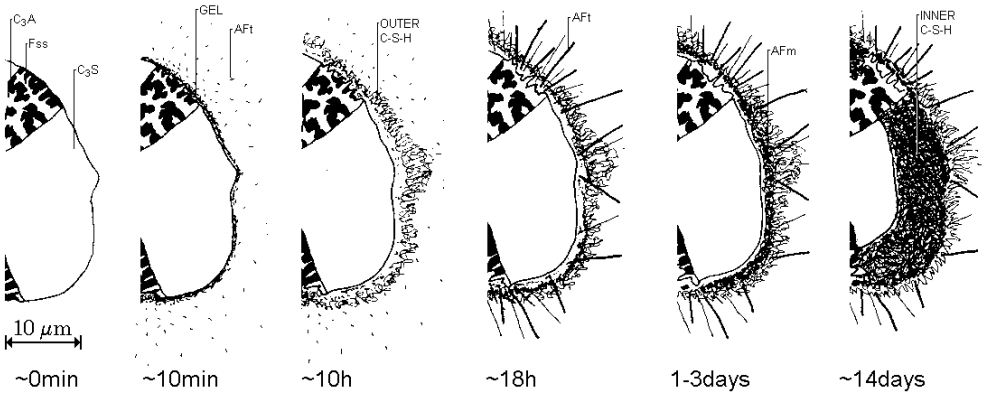


Figure 2.17: The condition of a cement particle at different times from water addition. The illustrations demonstrate how the hydrate (i.e. C-S-H, AFt and more) covers the unhydrated part of the cement particle (i.e. C_3S, C_2S, C_3A and C_4AF) as a function of time. The figure is from Scrivener [111]. See the center illustration of Figure 2.10 (Page 26) about the direct effect of hydrate cover (basically C-S-H-gel and AFt) on the shear viscosity η.

variety of ionic species into the solution between them. More precisely, the chemical process of interests are first of all, the ejection of ions, like of K^+, Na^+, SO_4^{-2}, Ca^{+2}, OH^- and SiO_4^{-4} into the solution, between the cement particles (see Figure 2.16). This gives the bulk solution a specific ionic strength I, defined with Equation 2.54. Also, as a result of this ion-ejection, the cement particles becomes coated with hydration products and gains a surface charge. Both negative and positive charges are present at the surface of a given cement particle [29]. However, depending on literature, the total charge is either reported as positive or as negative. For example, the total charge of circa +10 mV is reported by Edmeades et al. [29] and of −2 mV by Gustafsson [39].

2.5.2 Potential Energy V_T Between Cement Particles

Electrostatic Repulsive Potential Energy V_R Between Charged Particles

With both negative and positive charges at the surface of a cement particle, two such particles have a very strong tendency in coagulating. This is due to the electrostatic attraction between the opposite charged surfaces. This leads quickly to a very large coagulation state $J_t^{tot} = J_t^P + J_t$ (permanent and/or reversible), which results in a larger shear viscosity η as shown with Figure 2.11. However, with the introduction of negatively charged polymers that adsorbs to the positive charged surfaces, both the cement particles become negatively charged, resulting in their electrostatic repulsion and hence in reduced coagulation rate H (which results in a smaller J_t^{tot} value).

For the specific case shown in Figure 2.18, the adsorbed polymer is treated as a part of the solid surface[25]. Ions of opposite charge (counter-ions), like Ca^{+2}, are

[25]In this thesis, the solid surface is defined as the outer boundary of the hydrate products AFt and C-S-H (these products covers the unhydrated part of the cement particle, c.f. Section 2.5.1). This boundary includes the part of the polymer that is absorbed (or fused) into the hydrate, either by the hydrate "growth" or by adsorption into "canyons", like shown in Figure 2.20, to the left. The part of the adsorbed polymer that is not fused into the hydrate, is not a part of the solid surface. With this definition, the solid surface defines the outer boundary of the cement particle. As such, the distance between two approaching cement particles is defined as between their solid surfaces. This solid-surface-to-solid-surface distance is represented with D_s (see Figure 2.22, Page 45).

attracted toward the solid surface and ions of like charge (co-ions), like SO_4^{-2}, are repelled away from the solid surface [115]. This together with the mixing tendency of thermal motion, leads to the formation of **electric double-layer** [115]. The electric double-layer can be regarded as consisting of two regions. An inner region (of thickness δ) called the **Stern layer**, and a outer region (of effective thickness $1/\kappa$) called the **diffuse layer**. In the diffuse layer, ions are distributed according to the influence of electrical force and random thermal motion [115]. As shown in Figure 2.18, the thickness of Stern layer δ, is equal to the radius of ions that are adsorbed to the solid surface (namely, the Ca^{+2} ion for this specific case).

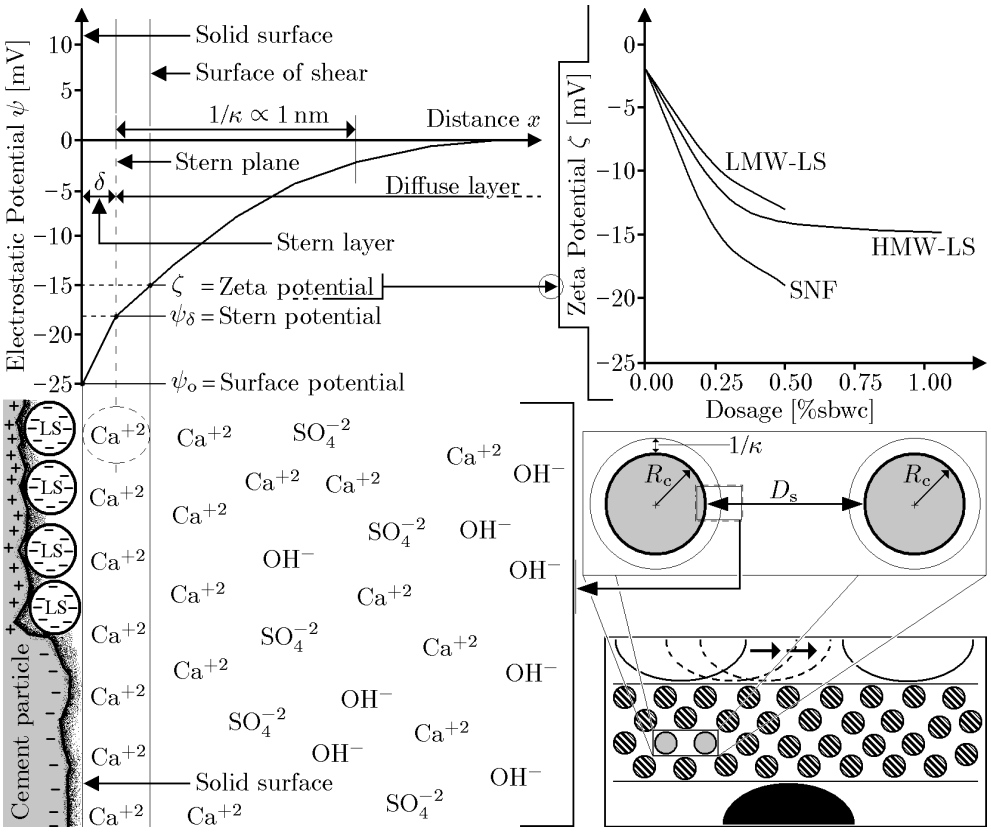


Figure 2.18: Top left and bottom: Schematic representation of the structure of the electric double-layer according to Stern's theory [115]. In this theory, the adsorbed lignosulfonate polymer (LS) is treated as a part of the solid surface. Its size drawn here, is severely underestimated for convenience. The correct diameter is about 10 nm [18]. Also, not all ions involved are shown. Top right: Zeta potential ζ as a function of polymer dosage in stirred OPC pastes ($w/c = 0.5$), 20 minutes after water additions [39, 18]. The markings "LMW", "HMW" and "SNF" are defined in Section 4.2.2. The bottom right illustration is extracted from Figure 2.11.

The change in electrostatic potential ψ in a charged medium, from a charged solid surface, can be calculated with one of the (inhomogeneous) Maxwell's equations, presented with Equation 2.51 [23, 117].

$$\nabla \cdot \mathbf{E} = \nabla \cdot (-\nabla\psi) = \rho_e/\epsilon \quad \Rightarrow \quad \nabla^2\psi = -\rho_e/(\epsilon_o\epsilon_r) \tag{2.51}$$

The above equation is frequently referred as Poisson equation [115, 52, 79], because

of its mathematical[26] form. The term $\mathbf{E}$ [V/m] is the electric field, and has similar functionality as gravity $\mathbf{g}$ [m/s^2] in the physics of mechanics. As mentioned previously, the term ψ [V] is designated as electrostatic potential. It plays a similar role as the geopotential Φ [m^2/s^2] in Equation B.26, Page 387 ($\mathbf{E} = -\nabla\psi \Leftrightarrow \mathbf{g} = -\nabla\Phi$). Finally, the terms ρ_e, ϵ_o and ϵ_r are the charge density [C/m^3], the permittivity of vacuum [F/m] and the relative permittivity [unit less], respectively.

With various simplifications and approximations, Equation 2.51 can be solved by analytical means for a single isolated charged (cement) particle. First, by applying the Gouy-Chapman approximations, followed by some improvement from the Stern's theory, an approximated electrostatic potential $\psi = \psi(x)$ results [115, 52, 79]. With the solution $\psi = \psi(x)$, one can calculate the electrostatic potential energy V [J] between the cement particle and an unit charge q with $V = q\,\psi(x)$ [92]. With $d\psi(x)/dx \geq 0\ \forall\ x$, as shown in Figure 2.18, an electrostatic repulsive potential energy $V_\mathrm{R} = -e\,\psi(x)$ results between a negative charged cement particle and a negative charged ion, like of the OH^-, pushing the ion away from the solid surface. The repulsive force that exits between them is given by $\mathbf{F} = -\nabla V_\mathrm{R} = -dV_\mathrm{R}/dx\,\mathbf{i}_\mathrm{x}$. Calculating the repulsive potential energy $V_\mathrm{R} = V_\mathrm{R}(\psi_1, \psi_2, \dots)$ that builds up between two charged cement particles 1 and 2 when their diffuse layers $(1/\kappa)$ start to overlap, is much more complicated. However, this is possible with some additional approximations [79]. For example, the repulsive potential energy between two spherical surfaces of the same radius R_c is given by [141]:

$$V_\mathrm{R} = 2\,\pi\,\epsilon_\mathrm{o}\,\epsilon_\mathrm{r}\,R_\mathrm{c}\,\psi_\delta^2\,\ln\left(1 + \exp\left(-\kappa\,D_\mathrm{s}\right)\right) \qquad (\kappa\,R_\mathrm{c} \gg 1) \tag{2.52}$$

The term ψ_δ is called the Stern potential, and is usually approximated with the zeta potential $\psi_\delta \approx \zeta$ (see Figure 2.18), because of the uncertainties and difficulties in measuring the former [79]. D_s is the solid-surface-to-solid-surface distance between the cement particles (see Footnote 25). The parameter κ is called the Debye-Hückel parameter, and is given by Equation 2.53 [52]. With this parameter, one can calculate the (effective) thickness of the diffuse layer with $1/\kappa$ [52, 79]. The four parameters ψ_δ, ζ, D_s and $1/\kappa$ are shown in Figure 2.18.

$$\kappa = \left[\frac{2000\,F_\mathrm{d}^2}{\epsilon_\mathrm{o}\epsilon_\mathrm{r}\,R\,T}\right]^{\frac{1}{2}}\sqrt{I} \qquad \text{in units of m}^{-1} \tag{2.53}$$

The term F_d in the above equation, is the Faraday's constant and I is the ionic strength (in the bulk solution, between the cement particles), and is given with [52]:

$$I = \frac{1}{2}\sum_\mathrm{i} c_\mathrm{i} z_\mathrm{i}^2 \tag{2.54}$$

The summation is over all the species of ion present in the bulk solution, between the cement particles [52]. The term c_i is the ionic concentration[27] in the bulk solution [mol/l] and z_i is the ion valence. When calculating the ionic strength I, the ionic consecration [mol/l] near and at the solid surface may not be included (see for example the textbook by Hunter [52]). The phrase "bulk solution" is used here, instead of just "solution", to highlight this.

[26] Mathematical function of the type $\nabla^2 u = f$ is generally referred as a Poisson equation [117].

[27] If c_i is represented with the unit of [mol/m^3] when calculating the ionic strength I, Equation 2.53 must be replaced with $\kappa = [(2\,F_\mathrm{d}^2)/(\epsilon_\mathrm{o}\epsilon_\mathrm{r}\,R\,T)]^{1/2}\sqrt{I}$.

Attractive Potential Energy V_A Between Particles

There is also an attractive force $\mathbf{F}_A = -\nabla V_A$ that exists between two cement particles. This force consists of three types of intermolecular attraction[28], namely the Keesom, the Debye and the London-dispersion forces [12, 79, 115]. These three forces are usually referred as the *van der Waals forces* [115, 79, 56]. A short description of these forces will now follow:

Keesom force: A force between two molecules with a permanent electric dipole moments. They mutually orientate each other in such way that ends of opposite sign are adjacent. In this orientation, the molecules strongly attract each other.

Debye force: A force which forms when molecule with permanent dipole induces a dipoles in the neighboring nonpolar molecule in such way that, on the average, attraction results between them.

London dispersion force: The electron distribution in a nonpolar molecule is symmetric on the average. However, the electrons themselves are in constant motion and at any given instant, one part of the molecule has an excess of them. When two nonpolar molecules are close enough, their fluctuation charge distribution tend to shift together, adjacent ends always having opposite sign and so always causing an attractive force.

Usually, it is the London dispersion force that is of most important to the overall van der Waals forces [115, 79]. For such a case, J. C. Hamaker (1937) derived an expression for the attractive potential energy V_A [115, 79] shown with Equation 2.55. This expression applies for two particles with radiuses r_1 and r_2, separated by the solid-surface-to-solid-surface distance D_s in vacuum [115, 79].

$$V_A = -\frac{A}{6}\left[\frac{2\,r_1\,r_2}{\tilde{s}-\tilde{x}} + \frac{2\,r_1\,r_2}{\tilde{s}-\tilde{y}} + \ln\left(\frac{\tilde{s}-\tilde{x}}{\tilde{s}-\tilde{y}}\right)\right] \tag{2.55}$$

The term A is the Hamaker constant and is usually in the range between 10^{-19} and 10^{-20} J [79]. The terms $\tilde{x}$, $\tilde{y}$ and $\tilde{s}$ are given with $\tilde{x} = (r_1 + r_2)^2$, $\tilde{y} = (r_1 - r_2)^2$ and $\tilde{s} = (r_1 + r_2 + D_s)^2$. The above equation can also apply between two particles submerged in a liquid dispersion medium (here, the solution between the cement particles). In doing so, the constant A must be replaced with the effective Hamaker constant A_{131} [115]. When one is considering a two equal sized cement particles $r_1 = r_2 = R_c$, the above equation becomes:

$$V_A = -\frac{A_{131}}{6}\left[\frac{2\,R_c^2}{\tilde{s}-\tilde{x}} + \frac{2\,R_c^2}{\tilde{s}} + \ln\left(\frac{\tilde{s}-\tilde{x}}{\tilde{s}}\right)\right] \tag{2.56}$$

where $\tilde{x} = 4R_c^2$ and $\tilde{s} = (2R_c + D_s)^2$. As the distance between two cement particles, approaches zero ($D_s \to 0$), the above energy potential approaches minus infinite ($V_A \to -\infty$). However, repulsion due to overlapping of electron clouds, named the Born repulsion, predominates at very small distances when the particles come into contact [115]. This means that the real total potential energy $V_T = V_R + V_A + V_S$, never goes to such extreme minimum as predicted by Equation 2.56 or by Equation 2.55.

The DLVO Theory: $V_T = V_R + V_A$

With the double-layer repulsive potential energy V_R (Equation 2.52) and the van der Waals attractive potential energy V_A (Equation 2.56) acting together, the total

[28] These three types of forces where postulated by van der Waals to explain non-ideal gas behavior.

potential energy $V_{\mathrm{T}} = V_{\mathrm{R}} + V_{\mathrm{A}}$ can be formed between two charged spherical particles of radius R_{c}. This result is a part of the well-known DLVO theory [115, 52, 79, 56]. The left illustration of Figure 2.19, demonstrates an example of total potential energy V_{T} between two cement particles[29] of radius $R_{\mathrm{c}} = 20\,\mu\mathrm{m}$. This radius is roughly equal to the radius of the cement particle shown in Figure 2.15 (diameter $\approx 40\,\mu\mathrm{m}$). Other values used are $\epsilon_{\mathrm{o}} = 8.854 \cdot 10^{-12}\,\mathrm{F/m}$, $\epsilon_{\mathrm{r}} = 78.5$ (water), $\psi_\delta \approx \zeta = -15\,\mathrm{mV}$ (see Figure 2.18), $T = 293.15\,\mathrm{K}$, $F_{\mathrm{d}} = 96486\,\mathrm{C/mol}$, $R = 8.3143\,\mathrm{J/(K \cdot mol)}$ and $A_{131} = 5.24 \cdot 10^{-22}\,\mathrm{J}$. The value of the Hamaker constant A_{131} used in this specific example, is extracted from Hattori [41]. The ionic strength I values used, are in the range what has been measured in cement pastes (see for example [78, 150]).

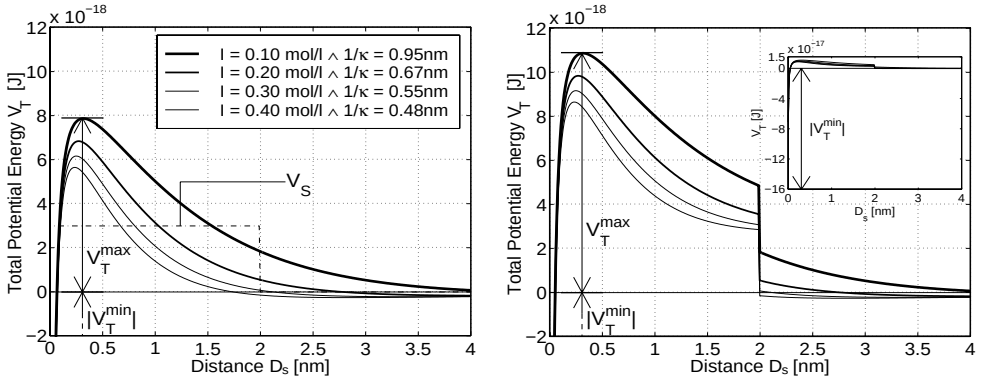


Figure 2.19: Total potential energy V_{T} without (to the left) and with (to the right) a steric effect V_{S}. The dashed dotted line in the left illustration demonstrates the value of V_{S} used.

As shown with the left illustration of Figure 2.19, the energy barrier $V_{\mathrm{T}}^{\max}$ against coagulation consist of $8 \cdot 10^{-18}\,\mathrm{J}$. This means that one of the cement particles only requires a speed[30] of $v = 3.9 \cdot 10^{-4}\,\mathrm{m/s}$ ($R_{\mathrm{c}} = 20\,\mu\mathrm{m}$) to overcome this barrier. However, when in a coagulated state, a much larger amount of kinetic energy is required for separating these two cement particles. This is due to how the energy minimum $|V_{\mathrm{T}}^{\min}|$ is much larger in magnitude, than the maximum value $V_{\mathrm{T}}^{\max}$. With smaller cement particles, repulsive potential energy V_{R} is much more effective in inhibiting a coagulation. For example with $R_{\mathrm{c}} = 1\,\mu\mathrm{m}$, the cement particles requires a speed of $v = 3.5\,\mathrm{cm/s}$ ($V_{\mathrm{T}}^{\max} = 3.9 \cdot 10^{-19}\,\mathrm{J}$) to make the coagulation a reality.

Traditionally, the particles that are influenced by the DLVO effects are considered to be of colloid particles. General definition of such particles is that at least one dimension is in the size range from 1 nm to 1 μm [115, 52, 79]. However, there is no clear distinction between the behavior of particles with somewhat larger dimensions than

[29] It should be clear that with the given approximations and assumptions made in forming Equations 2.52 and 2.56, the total potential energy V_{T} shown, is only an approximation of the actual potential energy that exists between two cement particles.

[30] $\frac{1}{2}mv^2 = V_{\mathrm{T}}^{\max} \Rightarrow v = [2\,(8 \cdot 10^{-18}\,\mathrm{J})/1.05 \cdot 10^{-10}\,\mathrm{kg}]^{1/2} = 3.9 \cdot 10^{-4}\,\mathrm{m/s}$; ($\rho = 3120\,\mathrm{kg/m^3}$). With this calculation it is assumed that coagulation should occur between two particles if they have sufficient mutual kinetic energy to surmount the potential energy barrier V_{T} that separates them. For perikinetic condition ($\dot{\gamma} = 0$), such a procedure can be shown to be inadequate because any particle having such a large amount of energy would lose it by frictional interactions with the dispersion medium (here, the solution between the cement particles) long before it is able to penetrate through the barrier [52]. However, under orthokinetic condition ($\dot{\gamma} \neq 0$), it can be argued that the loss of kinetic energy by this manner, is at least in part counterweighted by the gain due to the overall stirring of the suspension.

of the traditional colloidal particle [52]. As is shown with the example of the previous paragraph, cement particles as large as $R_c = 20\,\mu$m seem to be able to "behave" as a colloid particle, controlled by the action of the DLVO influence. However, when considering the pair of cement particles somewhat larger than $R_c = 20\,\mu$m, their kinetic energy (or inertia) starts to be sufficiently large in overcoming their mutual energy barrier V_T^{max} against coagulation, and their energy barrier $V_T^{max} + |V_T^{min}|$ against dispersion. When this condition applies, coagulation does not occur and the cement particles interact with each other rather by a pure hard sphere collision factor.

Steric Repulsive Potential Energy V_S

There is another potential energy effect that is also present, namely the steric repulsive potential energy V_S [J]. This effect basically consists of that the adsorbed plasticizing polymer (marked as "LS" in Figure 2.18) inhibits the cement particles to physically approach each other so closely as before [128]. The adsorbed skin of polymers thus forms a steric barrier against close contact. If this barrier is thicker than the distance to the potential energy minimum, the cement particles cannot approach each other close enough to stick [128]. In this manner, a coagulation is prevented [128]. The overall effect is usually referred as a steric[31] hindrance.

The dashed dotted line in the left illustration of Figure 2.19, demonstrates an example of V_S. The resulting total potential energy $V_T(D_s) = V_R(D_s) + V_A(D_s) + V_S(D_s)$ is shown in the right illustration. When plotting this last-mentioned illustration, it is assumed that the polymer contributing to steric hindrance, is without a surface charge (i.e. without an active anion), and that the polymer contribution to the zeta potential ζ is a part of the solid surface. Such condition is shown in the left illustration of Figure 2.20. In other words, if the potential energy $V_T(D_s) = V_R(D_s) + V_A(D_s) + V_S(D_s)$ plotted in the right illustration of Figure 2.19 is to be valid, the condition shown in the left illustration of Figure 2.20 must apply. This condition is not very realistic, since all lignosulfonate polymers contains surface charge and since they do not necessarily arrange themselves as conveniently as shown in this illustration.

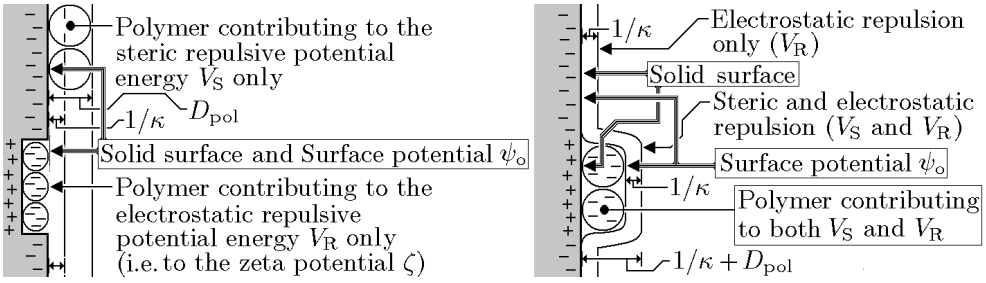


Figure 2.20: To the left: The condition that must apply if the right illustration of Figure 2.19, or illustration **B** in Figure 2.21, are to be correct. To the right: A more realistic presentation of adsorption. This configuration corresponds to illustration **C** in Figure 2.21. In this last mentioned illustration, the dashed line corresponds to electrostatic repulsion ($V_R + V_A$) and the solid line corresponds to steric and electrostatic repulsion ($V_R + V_A + V_S$).

Since the lignosulfonate polymer has a surface charge (consisting of SO_3^-, $C_6H_5O^-$ and COO^- groups), the surface potential ψ_o (see top left illustration of Figure 2.18)

[31] If present at relevant magnitude, the function V_S will also include the effect from the **hydration** (or **structural**) force. See [52, 56] about the (repulsive) hydration force.

is moved away from the solid surface to the outer boundary of the polymer. This is shown with the right illustration of Figure 2.20. With this, the electric double-layer (of thickness $\delta + 1/\kappa \approx 1/\kappa$) is also moved by the same distance. Therefore, if the thickness of the adsorbed polymers is designated with D_{pol}, the electrostatic repulsive potential energy $V_{\mathrm{R}}(D_{\mathrm{s}})$ must be shifted by the distance of $2D_{\mathrm{pol}}$. This is done by replacing $V_{\mathrm{R}}(D_{\mathrm{s}})$ with $V_{\mathrm{R}}(D_{\mathrm{s}} - 2D_{\mathrm{pol}})$. Hence, the total potential energy rather consists of $V_{\mathrm{T}}(D_{\mathrm{s}}) = V_{\mathrm{R}}(D_{\mathrm{s}} - 2D_{\mathrm{pol}}) + V_{\mathrm{A}}(D_{\mathrm{s}}) + V_{\mathrm{S}}(D_{\mathrm{s}})$. The reason for the shift of $2D_{\mathrm{pol}}$ instead of just D_{pol} will be explained shortly.

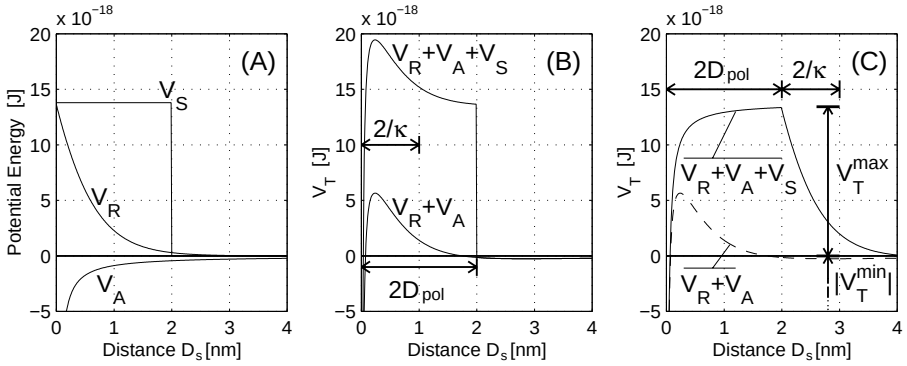


Figure 2.21: To the left: The potential energy contributions V_{R}, V_{A} and V_{S}. Center: $V_{\mathrm{T}}(D_{\mathrm{s}}) = V_{\mathrm{R}}(D_{\mathrm{s}}) + V_{\mathrm{A}}(D_{\mathrm{s}}) + V_{\mathrm{S}}(D_{\mathrm{s}})$. To the right: $V_{\mathrm{T}}(D_{\mathrm{s}}) = V_{\mathrm{R}}(D_{\mathrm{s}} - 2D_{\mathrm{pol}}) + V_{\mathrm{A}}(D_{\mathrm{s}}) + V_{\mathrm{S}}(D_{\mathrm{s}})$.

Illustration **A** in Figure 2.21 demonstrates the individual potential energy contributions V_{R}, V_{A} and V_{S}. These plots are generated by applying the values used in Figure 2.19, with the ionic strength of $I = 0.40\,\mathrm{mol/l}$. The steric potential used here, is larger than the one shown in Figure 2.19. Illustration **B** in Figure 2.21 demonstrates the resulting potential energy $V_{\mathrm{T}}(D_{\mathrm{s}}) = V_{\mathrm{R}}(D_{\mathrm{s}}) + V_{\mathrm{A}}(D_{\mathrm{s}}) + V_{\mathrm{S}}(D_{\mathrm{s}})$, while illustration **C** demonstrates plot of $V_{\mathrm{T}}(D_{\mathrm{s}}) = V_{\mathrm{R}}(D_{\mathrm{s}} - 2D_{\mathrm{pol}}) + V_{\mathrm{A}}(D_{\mathrm{s}}) + V_{\mathrm{S}}(D_{\mathrm{s}})$. The functional form of the plot shown in illustration **C**, does not necessarily have to be fully correct. Rather, the main objectives with it, is to demonstrate that the reach of the total potential energy V_{T} is not just $2D_{\mathrm{pol}}$, but rather $2D_{\mathrm{pol}} + 2/\kappa$.

The Reach of the Total Potential Energy $V_{\mathrm{T}} = V_{\mathrm{R}} + V_{\mathrm{A}} + V_{\mathrm{S}}$

As mentioned previously, the (effective) thickness of the diffuse layer at the solid surface is $1/\kappa$. This is also shown in the top left illustration of Figure 2.22. Because of diffuse layer overlapping, an electrostatic repulsion (V_{R}) starts to build up between two approaching cement particles, when their distance D_{s} becomes low as $2/\kappa$. Relative to this, it is possible to state that the reach of the repulsion V_{R} is $2/\kappa$.

As is mentioned in Section 6.4.1, a multi-layer polymer adsorption could be factual, rather than a mono-layer adsorption. This indicates that a negative charged polymer can also adsorb on a negative charged solid surface. Given the hydrophobic properties of the lignosulfonate core, then this is possible (see Section 4.2.2 about the structure of the lignosulfonate polymer). Such adsorption is shown in the right illustration of Figure 2.22. With this in mind, it is convenient to redefine the term D_{pol} to such that it describes the average thickness of the adsorbed polymer on the cement particles. In accordance with Footnote 25 (Page 38), the part of the polymers that are fused

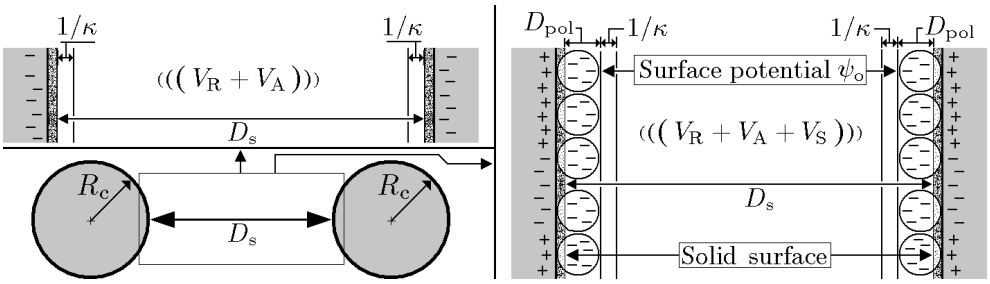


Figure 2.22: The top left illustration corresponds to the dashed line in illustration **C** of Figure 2.21. The right illustration corresponds to the solid line in illustration **C** of Figure 2.21.

into the solid surface, do not contribute to the thickness D_{pol}. This is indicated with the right illustration of Figure 2.22.

Consider two cement particles that are covered with polymers of (average) thickness D_{pol} and are approaching each other. As their distance D_{s} becomes lower than $2D_{\mathrm{pol}}$, a steric repulsive potential energy V_{S} starts to build up between them. Relative to this, it is possible to state that the reach of the repulsion V_{S} is $2D_{\mathrm{pol}}$.

By polymer adsorption, the surface potential ψ_{o} is moved by the distance D_{pol}, away from the solid surface. Hence, the electric double-layer (of thickness $\delta + 1/\kappa \approx 1/\kappa$) is moved by the same distance. Therefore, the reach of the electrostatic (V_{R}) and steric (V_{S}) repulsion can be considered as additive. As shown in the right illustration of Figure 2.22, this gives a reach of $2D_{\mathrm{pol}} + 2/\kappa$ for the total potential energy V_{T}.

2.5.3 Coagulation Rate H and Potential Energy V_{T}

The rate at which cement particles coagulate (permanently or reversibly), depends on the number of collisions per unit time, between them [115, 52, 79]. It also depends on the probability that their individual kinetic energy is sufficient to overcome the potential energy barrier $V_{\mathrm{T}}^{\mathrm{max}}$ that separates these particles [115, 52, 79]. As mentioned in Section 2.4.2, the Hattori-Izumi uses the perikinetic ($\dot{\gamma} = 0$) coagulation rate theory, established by Verwey and Overbeek [141, 42, 41]. This theory was presented with Equation 2.29 and is reproduced below.

$$-\frac{dn_{\mathrm{t}}}{dt} = \frac{H\, n_{\mathrm{t}}^2}{n_3} \tag{2.57}$$

The coagulation rate constant H in the above equation is given by [141, 42, 41]:

$$H = \frac{4\, K_{\mathrm{sm}}\, R_{\mathrm{c}}\, n_3}{(2/\kappa)\, \exp(V_{\mathrm{T}}^{\mathrm{max}}/k\,T)} \qquad \dot{\gamma} = 0 \ \wedge\ R_{\mathrm{c}} < 20\,\mu\mathrm{m} \tag{2.58}$$

The term k is the Boltzmann's constant and K_{sm} is the Smoluchowski's rapid coagulation rate constant, equal to $K_{\mathrm{sm}} = (4\,k\,T)/(3\,\eta_l)$, where the term η_l is the shear viscosity of the liquid dispersion medium (here, the solution between the cement particles). The total potential energy used in the above equation is of $V_{\mathrm{T}} = V_{\mathrm{R}} + V_{\mathrm{A}}$; i.e. the steric contribution is not taken into account for. To include the steric effects, an empirical modification of the above equation could be made. This could consist of replacing $(2/\kappa)$ with $(2/\kappa + 2D_{\mathrm{pol}})$, where D_{pol} is the pre-mentioned thickness of the adsorbed polymers. Also, the total potential energy V_{T} would have to consist of

something like what is shown in illustration **C** of Figure 2.21. The resulting modified coagulation rate constant H is given by Equation 2.59.

$$H = \frac{4\,f(K_{\mathrm{sm}}, R_{\mathrm{c}}, n_3)}{(2/\kappa + 2D_{\mathrm{pol}})\,\exp(V_{\mathrm{T}}^{\mathrm{max}}/k\,T)} \qquad \dot{\gamma} = 0 \;\wedge\; R_{\mathrm{c}} < 20\,\mu\mathrm{m} \tag{2.59}$$

The term f in the above equation, is some unknown function.

By using Equations 2.57 and 2.58 (or 2.59) in the theory of Section 2.4.2, it is assumed that the total number of reversible junctions J_{t} increases by a naturally occurring perikinetic coagulation only. In other words, it is assumed that the coagulation rate H exists only because of the Brownian motion of the cement particles. However, as discussed in a textbook by Hunter [52], the coagulation rate H is greatly influenced by the stirring of the suspension $\Rightarrow H = H(\dot{\gamma})$. The stirring causes the particles to be thrown together at a larger rate than the normal diffusion[32] rate, and hence the orthokinetic processes increases coagulation. An explanation of the problem is given by Smoluchowski and is reproduced in the pre-mentioned textbook [52]. There, it is shown that the orthokinetic process becomes increasingly important with increasing particle size. With larger particles, such as in emulsions (droplet diameter $\propto 10^{-7}\,\mathrm{m}$), orthokinetic coagulation can occur at up to as much as 10^4 times the perikinetic rate [115]. With particles at the lower end of the colloidal size range (diameter $\propto 10^{-9}\,\mathrm{m}$), stirring has relatively little effect on their rate of coagulation [115]. Since non-hydrated cement particles are in the size range from $1 \cdot 10^{-6}\,\mathrm{m}$ to $100 \cdot 10^{-6}\,\mathrm{m}$ [131] in diameter, the orthokinetic coagulation process plays an important role in determining the correct coagulation rate H and hence in determining the correct junction number J_{t}. As such, for cement particles, Equations 2.58 and 2.59 can only be valid under perikinetic condition. This is indicated by writing "$\dot{\gamma} = 0$" beside those two equations. Also written is "$R_{\mathrm{c}} < 20\,\mu\mathrm{m}$", indicating that those equations are not relevant for particles larger than $20\,\mu\mathrm{m}$ in "radius": As mentioned on Page 42, for particles with larger radius than $20\,\mu\mathrm{m}$, coagulation between two such particles does generally not occur. Rather, they interact with each other by a hard sphere collision factor alone.

In simulating experimental results by numerical means, the term "$H\,t$" is replaced with the memory module $\tilde{\Theta}$ in Chapter 9. The latter variable is defined with Equation 2.60 and is reproduced from Equation 9.4 (Page 210).

$$\tilde{\Theta}(t) = \int_0^t \beta(t - t')\; H(\dot{\gamma}, t')\; dt' \tag{2.60}$$

The term β is a memory function, defined in Section 9.3.1. After a large number of trials, the most successful coagulation rate function is retrieved. This function is shown in Equation 2.61 and is reproduced from Equation 9.10 (Page 213).

$$H(\dot{\gamma}, t) = \frac{K(t)}{\dot{\gamma}^2 + l} \qquad \dot{\gamma} \neq 0 \;\wedge\; R_{\mathrm{c}} \in [0.01,\, 20]\,\mu\mathrm{m} \tag{2.61}$$

The term l in the above equation, is an empirical constant, kept equal to $l = 1\,\mathrm{s}^{-2}$ at all times. This equation applies under orthokinetic condition, as indicated by writing "$\dot{\gamma} \neq 0$" beside it. Also written is "$R_{\mathrm{c}} \in [0.01,\, 20]\,\mu\mathrm{m}$", indicating the size domain of the particles, which this equation applies to, when orthokinetic condition is valid; i.e. for particles with somewhat lower dimension than $0.01\,\mu\mathrm{m}$, Equation 2.58 or Equation 2.59 start rather to apply (see previous discussion).

[32] Brownian motion is observed macroscopically as diffusion [79].

Ideally, Equation 2.61 should converge to Equation 2.58 or Equation 2.59, in the limit of $\dot{\gamma} \to 0$. Hence, it can be suggested that at least K (and possible also β) is dependent on the important variables R_c, $2/\kappa + 2D_{pol}$ and V_T^{max}, giving:

$$K = \frac{4\, f_1(K_{sm}, R_c, n_3, t, \dots)}{(2/\kappa + 2D_{pol})\, \exp(V_T^{max}/[kT + f_2(\dot{\gamma})])} \tag{2.62}$$

$$\beta = \beta(t - t', 2/\kappa + 2D_{pol}, V_T^{max}, \dots) \tag{2.63}$$

where f_1 and f_2 are some material functions. An example of total potential energy function V_T used in Equation 2.62, is shown in illustration **C** of Figure 2.21. In the limitation of the theoretical and experimental work done in this thesis, the above functions K and β are extracted by pure empirical means. This is apparent with Equations 9.6 and 9.11 (Page 210).

2.6 Summary

2.6.1 Section 2.2: The Governing Equation

In this part, the governing equation for the particle suspension is derived. In this derivation, both the suspended particles and the matrix are treated in the exactly the same manner, namely together as a group of solid particles. No distinction is made between the two phases. Examples of "fluids" that benefit from this derivation are the rings of Saturn, consisting of a solid ice and rock fragments (see Figures B.5 and B.6, Page 384), the sea ice floes in the Icelandic waters (see Figure 2.3, Page 18) and the fresh concrete flowing inside a formwork (see Figure 3.2, Page 54) or inside a viscometer (see Figure 3.8, Page 62). Flow of mortars and cement pastes inside viscometers are additional examples.

2.6.2 Section 2.3: The Constitutive Equation

In this part, the constitutive equation to be used in the following chapters is presented. It consist of $\boldsymbol{\sigma} = -p\,\mathbf{I} + \mathbf{T}$, where $\mathbf{T}$ is the extra stress tensor. Few other constitutive equations are also presented. In this thesis, the extra stress tensor is equal to $\mathbf{T} = 2\,\eta\,\dot{\boldsymbol{\varepsilon}}$. In Chapters 3, 5, 6, 8 and 10, the shear viscosity function η used in the extra stress tensor is the one for the Bingham model $\eta = \mu + \tau_o/\dot{\gamma}$. However, in Chapter 9, a more complicated shear viscosity model is used, as explained in Section 2.6.3.

2.6.3 Section 2.4 to Section 2.5: The Shear Viscosity

In a cement paste, the cement particles are continously colliding with each other. For each collision, a momentum is transferred from one cement particle to the next. With Equation 2.25 (Page 26), this momentum transfer is directly related to the shear viscosity η of the cement paste.

In Section 2.5.1, it was discussed how complex chemical reactions occur between the cement particles and water (generally designated as hydration). From Section 2.4.1, these reactions increase the above-mentioned momentum transfer and thus increase the shear viscosity η as a function of time. More precisely, the hydration will increase the shear viscosity η by (at least) three effects. These are by the increased phase volume Φ, surface roughness and coagulation state $J_t^{tot} = J_t^p + J_t$.

Phase Volume Φ and Surface Roughness (Direct Hydration Effect)

As hydration goes on with time, the following properties of the suspension are changed:

- **Phase Volume Φ:** With time, the cement particles become more and more coated with hydration products (namely C-S-H and AFt) as shown in Figure 2.17 (Page 38). This results in an increased "radius" R_c of all cement particles and in reduced free water (see Footnote 19, Page 25 about free water). In a more strict rheological terms, this means that the phase volume[33] Φ is increasing with time (however at different rate). This will result in a reduced maneuverability of the individual smaller cement particle, in avoiding the formation of a continuous bridge between the larger cement particles (see the center illustration of Figures 2.9 and 2.10, Page 25). Hence, with increased hydration, the value of $\tilde{N}_{ic}$ will increase, which again will give an increased shear viscosity η as shown with Equation 2.25 (Page 26).

- **Surface Roughness:** As the cement particles become more and more coated with hydration products, their surface roughness increases. This results in a better grip (or contact) between two colliding cement particles and hence gives a larger momentum transfer (Δp_{ic} and Δp_{dc}) between them during a (direct or indirect) collision. As shown with Equation 2.25, this results in an increased shear viscosity η.

Coagulation State $J_t^{tot} = J_t^p + J_t$ (Indirect Hydration Effect)

When looking at Equation 2.59, it is clear that the variables R_c, $2/\kappa + 2D_{pol}$ and V_T^{max} controls the coagulation rate H, when perikinetic condition apply (i.e. when $\dot{\gamma} = 0$). The same could be expected for the orthokinetic condition (i.e. when $\dot{\gamma} \neq 0$) as indicated with Equation 2.62. Because of the chemical reactions between cement particles and water, the variables R_c, $2/\kappa + 2D_{pol}$ and V_T^{max} will change with time. This will result in a different coagulation rate H as a function of time. More precisely, during hydration the following occurs:

- **Polymer Damage/Degrade:** With time, the cement particles become more and more coated with hydration products (namely C-S-H and AFt). This results in the following:

  - Increased "radius" R_c of all cement particles.
  - Possible decreased D_{pol}, since larger part of the polymers could be integrated (or fused) into the solid surface of the cement particle (see the right illustration of Figure 2.22, Page 45).

[33] According to Barnes et at. [9], the phase volume Φ is defined as fraction of space of the total suspension that is occupied by the suspended particles, or equally as $\Phi = V_p/(V_p + V_m)$. The term V_p is the volume of the suspended particles and V_m is the volume of the matrix. Total volume of the suspension is given as $V_t = V_p + V_m$ and is slightly decreasing with time due to the chemical shrinkage as mentioned in Footnote 16 (Page 21). Here, the suspended particles consist of the cement particles, and hence, the free water defines the matrix for this particular case (see Footnote 19, Page 25 about the free water). When the suspension consists of solid particles of a broad range in mass, dimension, shape and surface texture, with no distinct boundary to a matrix, the phase volume becomes a more relative term, defined by convenience. Here, the matrix is defined as the free water, while in Chapter 10, it is defined as all particles below 2 mm in diameter (see Footnote 1, Page 238).

- Other types of continuous degradation/damage of the adsorbing polymer could also be present as a direct or indirect result of hydration. This means that there could be a reduction in the steric repulsive potential energy V_S as a function of time. If so, this results in a diminished maximum value of the total potential energy V_T^{max} as a function of time.
- The zeta potential ζ ($\approx \psi_\delta$) could also be reduced because of the prementioned degrade/damage to the adsorbing polymer. This would mean that there is a reduction in the electrostatic repulsive potential energy V_R as a function of time (see Equation 2.52, Page 40). If so, this results in a further reduction in the maximum value of total potential energy V_T^{max} as a function of time.

- **Ionic Strength *I*:** As a part of the hydration process, ejection of ions like of K^+, Na^+, SO_4^{-2}, Ca^{+2}, OH^- and SiO_4^{-4} occurs into the solution, between the cement particles (see Figure 2.16, Page 36). This gives the bulk solution a specific ionic strength I that is changing with time. With increasing ionic strength I, as a function if time, the following occurs:
  - The (effective) thickness of the diffuse layer $1/\kappa$ will be reduced. This is apparent with Equation 2.53 (Page 40).
  - The electrostatic repulsive potential energy V_R will be diminished (Equations 2.52 and 2.53, Page 40). If so, this results in a diminished maximum value of the total potential energy V_T^{max} as a function of time. This is shown with the left illustration of Figure 2.19 (Page 42) where it is demonstrated how increased ionic strength, from $I = 0.1\,\text{mol/l}$ to $0.4\,\text{mol/l}$, results in a reduced energy barrier V_T^{max} (and reach $2/\kappa$).

From the above points, it is to be expected that the value of R_c is increasing, while the values of $2/\kappa + 2D_{pol}$ and V_T^{max} are decreasing with time (i.e. with hydration). As shown with Equations 2.59, 2.61 and 2.62, this results in an increased coagulation rate H, which again gives an increased coagulation state $J_t^{tot} = J_t^p + J_t$. The relationship between H and J_t^{tot} is perhaps more obvious under perikinetic condition ($\dot{\gamma} = 0$), by using the coagulation rate theory established by Verwey and Overbeek [141] (see also Equation 2.30, Page 30):

$$n_t^{tot} = \frac{n_3^{tot}}{H\,t + 1} \qquad (\dot{\gamma} = 0) \tag{2.64}$$

The term n_3^{tot} is the total number of (primary) cement particles, per unit volume, which can coagulate (i.e. "effective" total number of primary particles in the suspension). The term n_t^{tot} is their number[34] at later time $t > 0$. One should bear in mind from Page 42, that not all cement particles can coagulate. Therefore, the value of n_3^{tot} is not necessarily equal to the total number of cement particles present in the suspension (i.e. in the cement paste). From Equation 2.28 (Page 29), assuming that the condition $n_3^{tot} = n_t^{tot} + J_t^{tot}$ applies, Equation 2.64 can be rewritten to the following:

[34] The same type of relationship applies, between n_3^{tot} and n_t^{tot}, as between n_3 and n_t in Figure 2.12 (Page 29). It should be noted that n_3 and n_t are related to reversible coagulation only, while n_3^{tot} and n_t^{tot} are related to both reversible and permanent coagulation.

$$J_{\mathrm{t}}^{\mathrm{tot}} = n_3^{\mathrm{tot}} \left(\frac{H\,t}{H\,t+1} \right) = \overbrace{n_3^{\mathrm{p}} \left(\frac{H\,t}{H\,t+1} \right)}^{J_{\mathrm{t}}^{\mathrm{p}}} + \overbrace{\underbrace{(n_3^{\mathrm{tot}} - n_3^{\mathrm{p}})}_{n_3} \left(\frac{H\,t}{H\,t+1} \right)}^{J_{\mathrm{t}}} \tag{2.65}$$

As indicated with the latter part of the above[35] equation, the term n_3^{tot} can be written as $n_3^{\mathrm{tot}} = n_3^{\mathrm{p}} + (n_3^{\mathrm{tot}} - n_3^{\mathrm{p}}) = n_3^{\mathrm{p}} + n_3$, where the term n_3^{p} describes the total number of (primary) cement particles that can undergo a permanent coagulation. The remaining number $n_3^{\mathrm{tot}} - n_3^{\mathrm{p}} = n_3$ describes then the total number of (primary) cement particles that can undergo a reversible coagulation. The last part of the above equation, namely $J_{\mathrm{t}} = n_3\,H\,t/(H\,t+1)$, is exactly the same as Equation 2.31 (Page 30).

From Equation 2.65, it is clear that increased coagulation rate H results in a larger coagulation state $J_{\mathrm{t}}^{\mathrm{tot}} = J_{\mathrm{t}}^{\mathrm{p}} + J_{\mathrm{t}}$ at the given time t. In either case of permanent coagulation state $J_{\mathrm{t}}^{\mathrm{p}}$ or reversible coagulation state J_{t}, bridges of small cement particles will form more frequently between the larger ones, as shown[36] in illustrations **C** and **D** of Figure 2.11 (Page 27). Being simultaneously in contact with two large cement particles, such bridge acts as a conduit for momentum transfer between them. With the more frequent formation of bridges between the larger cement particles, a larger value of $\tilde{N}_{\mathrm{ic}}$ will result, which again results in an increased shear viscosity η. This is shown with Equation 2.25 (Page 26).

In Section 2.4.2, a theory about the bookkeeping of the number of reversible junctions J_{t} is presented. With some important modifications, this last-mentioned theory is used in the computational part of this thesis, or more precisely in Chapter 9. That two types of coagulation are occurring simultaneously, namely the permanent coagulation ($J_{\mathrm{t}}^{\mathrm{p}}$) and the reversible coagulation (J_{t}), does not pose any real problem in Chapter 9. The issue of how these two types of coagulation are dealt with separately, is explained in the latter part of Section 9.3.2.

Summarizing the above text: In Section 2.4.1, the shear viscosity η of cement paste, is directly related to momentum exchange (by direct and indirect collisions) between cement particles. With the chemical reactions between cement particles and water, this momentum exchange will increase in a special manner, as explained above. With increasing momentum exchange, an increased shear viscosity η will result.

[35] The variables $J_{\mathrm{t}}^{\mathrm{tot}}$, $J_{\mathrm{t}}^{\mathrm{p}}$, J_{t}, n_3^{tot}, n_3^{p} and n_3 have all the physical unit of $[\mathrm{m}^{-3}]$ for consistency with the theory presented in Section 2.4.2.

[36] See also the right illustration of Figures 2.9 and 2.10, Page 25.

Chapter 3

Coaxial Cylinders Viscometer

3.1 Introduction

The coaxial cylinders viscometers used in this work have a stationary inner cylinder and rotating outer cylinder. They are designated as the ConTec **BML** Viscometer 3 [145, 147] and ConTec Viscometer 4, and are shown in Figure 3.1. The former (with the geometry of: $(R_\mathrm{i}, R_\mathrm{o}, h) = (10, 14.5, 19.9)$ cm) is used when measuring fresh concrete, while the latter (with the geometry of: $(R_\mathrm{i}, R_\mathrm{o}, h) = (8.5, 10.1, 11.6)$ cm) is used when measuring fresh mortar and cement paste. When rheological measurements are conducted on a Bingham fluid with either device, then by measuring the applied torque on the inner cylinder at the different angular velocities ω_o, one can connect these measured values with one straight line. From its slope H and its point of intersection with the ordinate G, the plastic viscosity μ and the yield value τ_o of the tested material are calculated. When representing the rheological results of this thesis, the parameters τ_o and μ are used in favor to the G and H values. The reason is that the latter two are not solely dependent on the flow characteristics of the test material, but rather also dependent on the geometry of the viscometer.

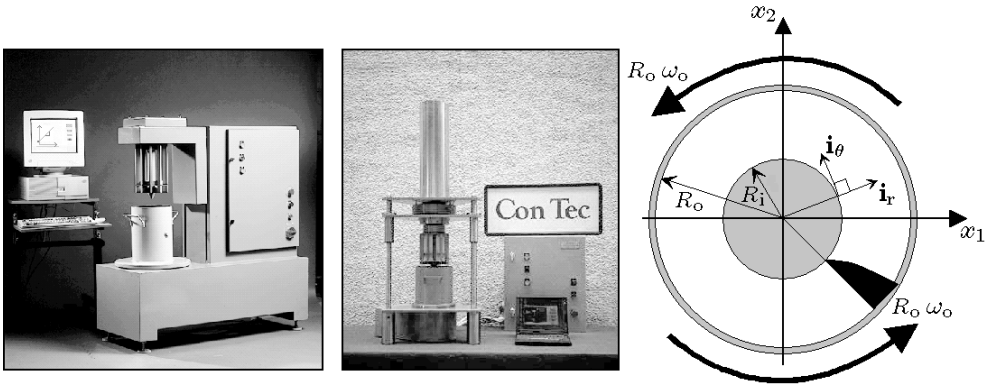


Figure 3.1: The ConTec **BML** Viscometer 3 (to the left) and ConTec Viscometer 4 (center). To the right is a schematic top view of both viscometers.

The objectives of this chapter are as follows: **1**) Section 3.2: Classifying a solid state (i.e. a plug state) and a viscoplastic state inside a **viscoplastic material**. The result produced are later used in Sections 3.5.2 and 7.9, and also in some places in

Chapters 9 and 10. With their yield value properties, the fresh concrete, mortar and cement paste are classified as a viscoplastic fluid (see for example the textbook by Malvern about viscoplastic fluids [72]).

2) Section 3.3: Demonstrating the underlying physics when converting measured data H and G to the material properties μ and τ_o. The equation used in this conversion is known as the Reiner-Riwlin equation. The result of this equation is applied in Chapters 5, 6 and 10.

3) Section 3.4: Explaining how the measured data H and G are produced by the viscometers. Depending on the viscometer and on the type of test material, different approaches are applied. This information is used in Chapters 5, 6 and 10.

4) Section 3.5: Constructing an estimate of the error generated when plug (i.e. solid state) is occurring inside the test material, during a measurement. The background of this part is related to the concern for some of the mortar mixes in this thesis, which had a rather high ratios of yield value to plastic viscosity, τ_o/μ. Such materials are prone to form a plug during the viscometric measurement. The conclusions of this section is (implicitly) used in Chapters 5 and 6.

3.2 Viscoplastic Fluid

3.2.1 von Mises Yield Condition

The viscoplastic material can generally be divided into two domains. One domain Ω_p is in a viscoplastic state, while the other domain Ω_e is in a solid state[1]. When in a viscoplastic state, the continuum behaves as a viscous fluid. When in a solid state, the continuum behaves as a rigid body (from a fluid perspective). Here, it will be assumed that this solid state corresponds to Hooke's law for an isotropic continuum, described with Equation 3.1. A more detailed or *accurate* description of the solid state is not necessary, since it can also be described with a rigid body motion (something which is only needed from the perspective of fluid flow). Therefore, for the sake of simplicity, Hooke's law is assumed to be sufficient description.

$$\boldsymbol{\sigma}^{E} = \tilde{\lambda}_e \, \mathrm{tr}(\boldsymbol{\varepsilon}) \, \mathbf{I} + 2\tilde{\mu}_e \, \boldsymbol{\varepsilon} \quad \forall \quad \mathbf{x} \in \Omega_e \tag{3.1}$$

The parameters $\tilde{\lambda}_e$ and $\tilde{\mu}_e$ are the Lamé constants of elasticity [74]. The term $\boldsymbol{\varepsilon}$ is the (Eulerian infinitesimal) strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{T}\right)$, and $\mathbf{I}$ is the unit dyadic. The term $\mathbf{u}$ is known as the **displacement vector** [74] and is given by $\mathbf{u} = \mathbf{x} - \mathbf{X}$ (see Equation 2.8, Page 14 about $\mathbf{x}$ and $\mathbf{X}$). For the viscoplastic state, the constitutive equation will be described with Equation 2.23 (Page 17), reproduced below.

$$\boldsymbol{\sigma}^{P} = -p\,\mathbf{I} + \mathbf{T} \quad \wedge \quad \mathbf{T} = 2\,\eta\,\dot{\boldsymbol{\varepsilon}} \quad \forall \quad \mathbf{x} \in \Omega_p \tag{3.2}$$

As indicated with the two above equations, the zone that is in a solid state is always designated with Ω_e, while the region of viscoplastic state is designated by Ω_p. Consequently, the symbol $\Omega \in [\Omega_e \cup \Omega_p]$ represents the whole continuum of interest.

Earlier studies report that the flow behavior of fresh concrete, mortar and cement paste can be closely approximated with a Bingham model [28, 128]. Using this model,

[1] In this thesis the terms **solid state** and **consolidated state** have a different meaning. The latter starts to appear in the end of the dormant period because of hydration of the cement clinker (see Reactions 2.49 and 2.50, Page 37), whereas the former applies whenever the shear rate is zero $\dot{\gamma} = 0$ (see Equation 3.5).

the shear viscosity function η will have the following form:

$$\eta = \mu + \frac{\tau_o}{\dot{\gamma}} \tag{3.3}$$

where the shear rate $\dot{\gamma}$ is given by Equation 2.24 (Page 18). As mentioned in Section 2.3, the terms τ_o and μ are known as the **yield value** [128] (or yield stress [9]) and **plastic viscosity** [9, 128], respectively. In this chapter, the yield value will also be refereed to as the *dynamic yield value*, to distinguish from the *static yield value*.

The **deviator stress tensor** is defined with: $\mathbf{S} = \boldsymbol{\sigma} - (\mathrm{tr}(\boldsymbol{\sigma})/3)\,\mathbf{I}$ [74]. One of its characteristics is that[2] $\mathrm{tr}(\mathbf{S})$=0. This tensor can be used to distinguish between the solid state ($\boldsymbol{\sigma}^{\mathrm{E}}$) and the viscoplastic state ($\boldsymbol{\sigma}^{\mathrm{P}}$) by using the von Mises yield condition, as is shown in the following equation:

$$\boldsymbol{\sigma} = \begin{cases} \boldsymbol{\sigma}^{\mathrm{P}} & \text{if} \quad (\mathbf{S}^{\mathrm{E}} : \mathbf{S}^{\mathrm{E}})/2 \geq \tau_y^2 \\ \boldsymbol{\sigma}^{\mathrm{E}} & \text{if} \quad (\mathbf{S}^{\mathrm{P}} : \mathbf{S}^{\mathrm{P}})/2 \leq \tau_o^2 \end{cases} \tag{3.4}$$

The terms τ_y and τ_o are the **static** and the **dynamic yield value**, to be explained shortly. The von Mises yield condition is usually written[3] as: $-\mathrm{II}_{\mathbf{S}} = C_y$ [74], where $\mathrm{II}_{\mathbf{S}} = (\mathrm{tr}(\mathbf{S})\,\mathrm{tr}(\mathbf{S}) - \mathbf{S} : \mathbf{S})/2 = -(\mathbf{S} : \mathbf{S})/2$ is the **second deviator stress invariant** and C_y is the **yield constant**, equal to either τ_y^2 or τ_o^2 in this thesis.

According to Equation B.21 (Page 387), the condition $\mathrm{tr}(\dot{\boldsymbol{\varepsilon}})$= 0 applies due to the incompressibility of the cement based material and hence $\mathrm{tr}(\mathbf{T}) = 2\,\eta\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}) = 0$. This means that the extra stress tensor $\mathbf{T}$ is also a deviator stress tensor[4]: $\mathbf{S}^{\mathrm{P}} = \mathbf{T}$. Since[5] $(\mathbf{T} : \mathbf{T})/2 = (\mu\,\dot{\gamma} + \tau_o)^2$, the condition for a solid state $(\mathbf{T} : \mathbf{T})/2 \leq \tau_o^2$, can equally be replaced with[6] $\dot{\gamma} = 0$. Also, by recognizing the term $(\mathbf{T} : \mathbf{T})/2 = \tau^2$ as the *von Mises shear stress*, the condition $\tau \leq \tau_o$ can also be used. To summarize, the transition **from** a viscoplastic state **to** a solid state (also known as plug state) has taken place when the equivalent conditions becomes valid:

$$(\mathbf{T} : \mathbf{T})/2 \leq \tau_o^2 \quad \Leftrightarrow \quad \dot{\gamma} = 0 \quad \Leftrightarrow \quad \tau \leq \tau_o \tag{3.5}$$

3.2.2 The Static and the Dynamic Yield Value

There are two types of yield values that can be related to concrete, mortar and cement paste, namely the static yield value τ_y and the dynamic yield value τ_o [145, 40]. The former is related to the amount of shear necessary to make the test material start flowing (i.e. the transformation from a solid state to a viscoplastic state), and the latter is more responsible for stopping a flowing test material (i.e. the transformation from a viscoplastic state, back to the solid state).

In this thesis, the main focus will be on the dynamic yield value τ_o rather than the static one τ_y. This is because of how the former value is more responsible for stopping the flowing concrete inside the mold or formwork (see Figure 3.2). The static yield value τ_y is also important, for example in determining the pressure exerted by the fresh concrete on the formwork (see Section 9.9.4). In the limitation of the experimental work done here, the static yield value is only measured for the cement paste, then

[2] $\mathrm{tr}(\mathbf{S}) = \mathrm{tr}(\boldsymbol{\sigma}) - (\mathrm{tr}(\boldsymbol{\sigma})/3)\,\mathrm{tr}(\mathbf{I}) = 0$ since $\mathrm{tr}(\mathbf{I})$=3.

[3] In some literature, this invariant is defined as $\mathrm{II}_{\mathbf{S}} = -(\mathrm{tr}(\mathbf{S})\,\mathrm{tr}(\mathbf{S}) - \mathbf{S} : \mathbf{S})/2$, and correspondingly the von Mises yield condition then becomes $\mathrm{II}_{\mathbf{S}} = C_y$ [72].

[4] $\mathbf{S}^{\mathrm{P}} = \boldsymbol{\sigma}^{\mathrm{P}} - (\mathrm{tr}(\boldsymbol{\sigma}^{\mathrm{P}})/3)\,\mathbf{I} = (-p\,\mathbf{I} + \mathbf{T}) - ([-p\,3 + \mathrm{tr}(\mathbf{T})]/3)\,\mathbf{I} = \mathbf{T}$

[5] $(\mathbf{T} : \mathbf{T})/2 = (2\,\eta)^2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}/2 = \eta^2\,(2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}) = (\mu + \tau_o/\dot{\gamma})^2\,\dot{\gamma}^2 = (\mu\,\dot{\gamma} + \tau_o)^2$

[6] Because the condition $\dot{\gamma} < 0$ does not exist, one cannot write $\dot{\gamma} \leq 0$.

Figure 3.2: Different types of casting of the fresh concrete [24]. Here it is the dynamic yield value τ_o that is more responsible for stopping the flowing concrete inside the formwork.

given as $\tau_y = \tau_o + \tilde{\tau}_o$ (see Equation 9.9 on Page 211). That is, the difference $\tau_y - \tau_o = \tilde{\tau}_o$ is related to thixotropic rebuild, as a consequence of (reversible) coagulation of the cement particles (see Section 9.3 about the reversible coagulation).

The difference $\tau_y - \tau_o$ can also be related to the additional resistance, because of the concomitant change in particle packing, when going from zero rate of deformation state $\dot{\boldsymbol{\varepsilon}} = 0$ (or $\dot{\gamma} = 0$) to non-zero rate of deformation state $\dot{\boldsymbol{\varepsilon}} \neq 0$ (or $\dot{\gamma} > 0$). This transformation is shown with the step from condition **A** to **B** in Figure 3.3. If the term $\tilde{\tau}_d$ designates the additional shear resistance created during this step, the static yield value can be written as $\tau_y = \tau_o + [\tilde{\tau}_o + \tilde{\tau}_d]$.

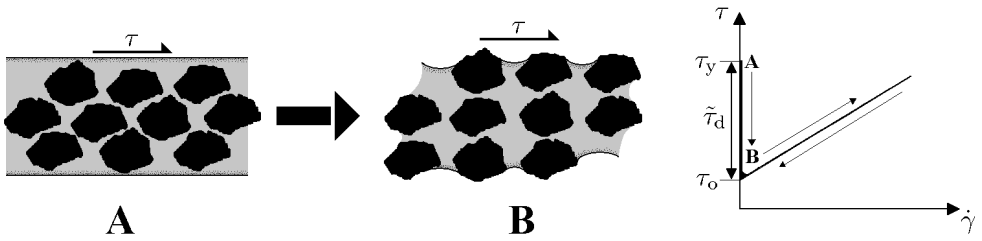


Figure 3.3: Condition **A** demonstrates a closed packed configuration and condition **B** an open one. The figure to the right demonstrates the corresponding shear stress as a function of shear rate $\dot{\gamma}$ [145] (assuming that Bingham model applies $\tau = \mu\,\dot{\gamma} + \tau_o$, when in viscoplastic state).

If the concrete batch consist of sufficient amount of binder (see the definition of binder in Footnote 2, Page 77) that surrounds the aggregates, the condition **B** will always prevail, meaning that $\tilde{\tau}_d = 0$. In addition to this, if the thixotropic yield value is at minimum ($\tilde{\tau}_o = 0$) because of a recent agitation, the static yield value will be in principle equal to the dynamic one $\tau_y - \tau_o = [\tilde{\tau}_o + \tilde{\tau}_d] = 0$. Accordingly, Equation 3.6 can then be written into the following form:

$$\boldsymbol{\sigma} = \begin{cases} \boldsymbol{\sigma}^{\mathrm{P}} & \text{if} \quad (\mathbf{S}^{\mathrm{E}} : \mathbf{S}^{\mathrm{E}})/2 \geq \tau_o^2 \\ \boldsymbol{\sigma}^{\mathrm{E}} & \text{if} \quad (\mathbf{S}^{\mathrm{P}} : \mathbf{S}^{\mathrm{P}})/2 < \tau_o^2 \end{cases} \tag{3.6}$$

According to Tanigawa and coworkers [121], for a relatively dry mixed fresh concrete (the condition **A** in Figure 3.3), the onset of viscoplastic state $(\mathbf{S}^{\mathrm{E}} : \mathbf{S}^{\mathrm{E}})/2 \geq \tau_y^2$ becomes quite dependent on the stress state $\boldsymbol{\sigma}$ (meaning $\tau_y = \tau_y(\boldsymbol{\sigma})$), contrary to for the onset of solid state $(\mathbf{S}^{\mathrm{P}} : \mathbf{S}^{\mathrm{P}})/2 \leq \tau_o^2 = \text{constant}$. A equation for the static yield value that describes this kind of dependency, can be given by the yield condition of Drucker and Prager [121, 78]: $\tau_y = k - \alpha\,\mathrm{I}_\sigma$, where $\mathrm{I}_\sigma = \mathrm{tr}(\boldsymbol{\sigma})$ is the first invariant

of the stress tensor $\boldsymbol{\sigma}$ (see for example [74]). The variables α and k are given by $\alpha = \tan(\Phi)/\sqrt{9 + 12\,\tan^2(\Phi)}$ and $k = 3C/\sqrt{9 + 12\,\tan^2(\Phi)}$. The term C is known as the cohesive strength and Φ is the angle of internal friction [99, 121]. To determine the onset of a viscoplastic state, the yield condition of Drucker and Prager is put into Equation 3.4. That is by writing $-\mathrm{II}_{\mathbf{S}}^{\mathrm{E}} = (\mathbf{S}^{\mathrm{E}} : \mathbf{S}^{\mathrm{E}})/2 = \tau_{\mathrm{y}}^2 = (k - \alpha\,\mathrm{I}_\sigma)^2$ and then rearranging, the following is produced:

$$-\mathrm{II}_{\mathbf{S}} = \tau_{\mathrm{y}}^2 \quad \Rightarrow \quad F = \sqrt{-\mathrm{II}_{\mathbf{S}}} \pm (\alpha\,\mathrm{I}_\sigma - k) = 0 \tag{3.7}$$

The latter part of Equation 3.7 might be a more familiar presentation of the yield condition by Drucker and Prager. When $F \geq 0$, the onset of viscoplastic state occurs. However, if the condition $F < 0$ prevails, the solid state will be present.

For the experimental results shown in Chapters 3, 5 and 6, thixotropic contribution is minimized ($\tilde{\tau}_{\mathrm{o}} \approx 0$). Also, since the *stepwise decreasing shear rate sequence* [67] is applied (see Figure 3.6), the condition **B** in Figure 3.3 is present at the beginning of an experiment ($\tilde{\tau}_{\mathrm{d}} \approx 0$). Hence, it is only the dynamic yield value τ_{o} that is measured for concrete and mortar. In Chapter 9, both τ_{o} and $\tilde{\tau}_{\mathrm{o}}$ are measured for cement paste.

3.3 Reiner-Riwlin Equation

The objectives of this part is to demonstrate the underlying physics when converting H and G into μ and τ_{o}, respectively. The equation used in this conversion is known as the Reiner-Riwlin equation [100]. As will be demonstrated, it can be used for both ConTec viscometers. In either case, it is the outer cylinder (with radius R_{o}) that rotates at different angular velocities ω_{o}, while the inner cylinder (with radius R_{i}) is stationary and registers the applied torque $\hat{T}$ from the test material.

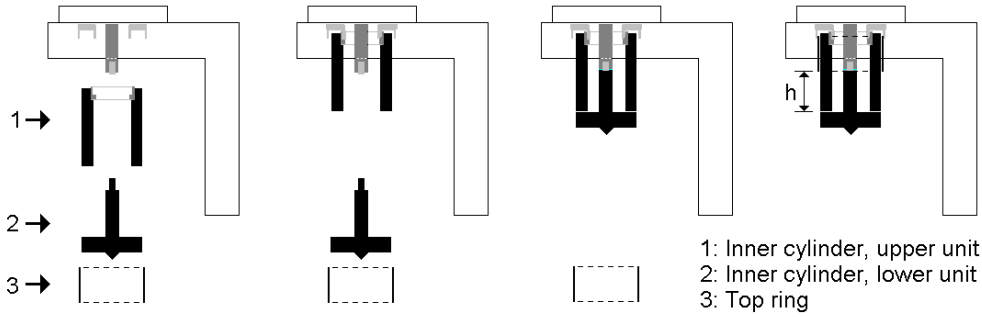


Figure 3.4: The assembly of inner cylinder to the ConTec **BML** Viscometer 3 [34]. The same configuration is used for the ConTec Viscometer 4.

As shown in Figure 3.4, the inner cylinder consists of three parts; the upper unit (measuring unit), the lower unit and the top-ring. It is only the upper unit that measures torque. The function of the lower unit is to eliminate the influence of shearing from the bottom plate of the outer cylinder (see Figure 3.5, in the middle). In this way, height independence can be assumed in the velocity function. The function of the top ring is somewhat less important, since its main objectives is to keep a constant height h where torque is measured, as shown in Figure 3.4, to the right. When using the ConTec Viscometer 4 (i.e. when doing a measurement on mortar or cement paste), the top ring is not used. The reason for this is explained in Section 5.3.3. Consequently, the height h is manually measured for each viscometric test.

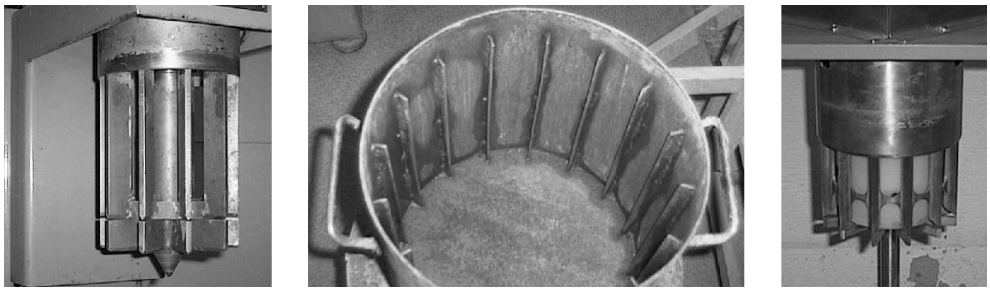


Figure 3.5: Both the inner (to the left) and outer cylinder (center) of ConTec **BML** Viscometer 3, consists of ribs (or vanes) aligned as a cylinder. As shown, the outer cylinder consists of a bucket. Same system is used for ConTec Viscometer 4 (to the right).

As demonstrated in Figure 3.5, the inner and outer cylinder consists of ribs aligned as a cylinder. In this way, it is the test material that will form the cylinders[7]. This leads to a larger cohesion (or stickiness) between the two cylinders and the test material in-between. This configuration reduces the danger for slippage.

3.3.1 Velocity Profile

For convenience, cylindrical coordinates will be used here. By using the general velocity field $\mathbf{v} = v_{\mathrm{r}}(r,\theta,z,t)\,\mathbf{i}_{\mathrm{r}} + v_{\theta}(r,\theta,z,t)\,\mathbf{i}_{\theta} + v_{\mathrm{z}}(r,\theta,z,t)\,\mathbf{i}_{\mathrm{z}}$, it is only possible to obtain solution by numerical means. However, some reasonable assumptions about the flow can be made, which makes an analytical approach possible:

1. With low Reynolds number (i.e. with low speed and high shear viscosity η) the flow is stable[8] and it is possible to assume a flow symmetry around the z-axis: $\mathbf{v} = v_{\mathrm{r}}(r,\theta,z,t)\,\mathbf{i}_{\mathrm{r}} + v_{\theta}(r,\theta,z,t)\,\mathbf{i}_{\theta}$.

2. With the lower unit of the inner cylinder (see Figure 3.4), the effect of shearing from the bottom plate is eliminated. Therefore a height independence can be assumed in the velocity function: $\mathbf{v} = v_{\mathrm{r}}(r,\theta,t)\,\mathbf{i}_{\mathrm{r}} + v_{\theta}(r,\theta,t)\,\mathbf{i}_{\theta}$. For both ConTec viscometers, this assumption is verified by numerical means in Chapter 8 (see Figure 8.16, Page 197 and Figure 8.21, Page 200).

3. Due to the circular geometry of a coaxial cylinders viscometer (see Figure 3.1) it is reasonable to assume pure circular flow with θ-independence. Also with time independence at each discrete angular velocity ω_{o}, the final velocity profile is given by:

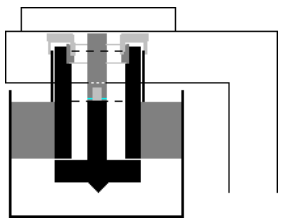
$$\mathbf{v} = v_{\theta}(r)\,\mathbf{i}_{\theta} \tag{3.8}$$

A physical description of velocity $\mathbf{v}$ is given by Equation 2.6 on Page 13 (see also the discussion below Equation 2.6).

[7] This is possible due to its Bingham properties. That is, between the ribs, the condition $\dot{\gamma} = 0$ applies and hence according to Equation 3.5, a solid state exists there.

[8] As random motion (Brownian motion) of water molecules (in water) is not understood as *fluid mechanical* turbulence, then neither are the random and spontaneous velocity contributions of individual solid particles (in suspension) understood as such. Rather, when groups of continuum particles (CPs), inside the continuum, start to travel coherently in circular paths, as a part of eddies or vortices of varying size, it is possible to define turbulence. More precisely, turbulent flow is characterized by a mixing action caused by eddies of varying size, throughout the continuum [106].

From here on, it is assumed that the outer cylinder rotates counterclockwise, giving $v_\theta(r) \geq 0$. This is done to simplify the calculations that follow. For both ConTec viscometers involved, the outer cylinder actually rotates clockwise. Nevertheless, the calculations apply fully for either clockwise or counterclockwise rotating outer cylinder. In addition, all the results that now follow apply only to the region in the gray area, shown in the right illustration.



3.3.2 Shear Stress and Torque

From Equation 3.8 the velocity gradient tensor $\nabla\mathbf{v}$ can be calculated, which through Equation 2.20 (Page 17), gives the strain rate tensor:

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r}\right)(\mathbf{i}_\mathrm{r}\mathbf{i}_\theta + \mathbf{i}_\theta\mathbf{i}_\mathrm{r}) \tag{3.9}$$

Substituting the result from Equation 3.9 into Equation 2.24 (Page 18) gives the shear rate that applies inside the test material.

$$\dot{\gamma} = \left|\frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r}\right| \tag{3.10}$$

Combining Equations 3.2 and 3.9, yields the extra stress tensor:

$$\mathbf{T} = \eta(\dot{\gamma})\left(\frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r}\right)(\mathbf{i}_\mathrm{r}\mathbf{i}_\theta + \mathbf{i}_\theta\mathbf{i}_\mathrm{r}) \tag{3.11}$$

The term $T_{r\theta}$ is now extracted directly from Equation 3.11:

$$T_{r\theta}(r) = \eta(\dot{\gamma})\left(\frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r}\right) \quad \forall\ r \in [R_\mathrm{i}, R_\mathrm{s}] \tag{3.12}$$

The term R_s in the above, designates the boundary between the viscoplastic zone Ω_p and the solid zone Ω_e. Generally the condition $R_\mathrm{i} < R_\mathrm{s} < R_\mathrm{o}$ applies. However when no plug is occurring inside the test material, the condition $R_\mathrm{s} = R_\mathrm{o}$ becomes valid. Now, consider the test material (inside the coaxial cylinders viscometer) to be divided into cylindrical shells, each with the thickness δR. The equation of rotational motion for each such shell is written as [73]:

$$I_\mathrm{R}\frac{d\boldsymbol{\omega}(r,t)}{dt} = \hat{\mathbf{T}} = \hat{T}(r+\delta R/2, t)\,\mathbf{i}_\mathrm{z} - \hat{T}(r-\delta R/2, t)\,\mathbf{i}_\mathrm{z} \tag{3.13}$$

where the term I_R is the moment of inertia of the specific shell. The relationship between angular velocity ω and velocity v_θ is $\boldsymbol{\omega} = \mathbf{i}_r \times v_\theta(r,t)\,\mathbf{i}_\theta/r = v_\theta(r,t)/r\ \mathbf{i}_\mathrm{z}$. The term $\hat{\mathbf{T}}$ designates the sum of external torques applied to the cylindrical shell, from its surroundings. The torque term $\hat{T}(r+\delta R/2, t)\,\mathbf{i}_\mathrm{z}$ accelerates the rotational motion of the cylindrical shell, because of positive work done by the outer and rotating cylinder (which is driven by an engine). On the other hand, the torque term $-\hat{T}(r-\delta R/2, t)\,\mathbf{i}_\mathrm{z}$ slows down this shell because of negative work from the inner and stationary cylinder. With the steady state $v_\theta = v_\theta(r) \Rightarrow d\omega/dt = 0$, the above equation becomes:

$$\hat{T}(r + \frac{\delta R}{2}) = \hat{T}(r - \frac{\delta R}{2}) \tag{3.14}$$

Equation 3.14 applies at the arbitrary distance r, anywhere inside the test material. Therefore, as long as there is time independence for the given measurement, the torque is a constant inside the test material.

$$\hat{T} = \text{constant} \quad \forall \; r \in [R_\text{i}, R_\text{o}] \tag{3.15}$$

The above result applies regardless if plug is occurring or not.

The torque applied **on** a cylindrical shell **from** its outer material, can be calculated with the help of Cauchy's stress principle [72]:

$$\hat{\mathbf{T}} = \hat{T}\,\mathbf{i}_\text{z} = \int_0^h \int_0^{2\,\pi} r\,\mathbf{i}_\text{r} \times (\mathbf{i}_\text{r} \cdot \boldsymbol{\sigma}(r)\, r\, d\theta\, dz) = 2\,\pi\, r^2\, h\, T_{r\theta}(r)\,\mathbf{i}_\text{z} \tag{3.16}$$

The term h is the height of the cylindrical shell. This term is for example shown in Figure 3.4. Using the above result in calculating the torque applied **from** the test material **on** the inner cylinder, gives:

$$\hat{T} = 2\,\pi\, R_\text{i}^2\, h\, T_{r\theta}(R_\text{i}) \tag{3.17}$$

Solving for $T_{r\theta}(r)$ in Equation 3.16 and then combining the result with Equation 3.12, produces Equation 3.18.

$$\eta(\dot{\gamma}) \left(\frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r} \right) = T_{r\theta}(r) = \frac{\hat{T}}{2\,\pi\, r^2\, h} \quad \forall \; r \in [R_\text{i}, R_\text{s}] \tag{3.18}$$

3.3.3 Shear Rate

Since it is the test material that is trying to get the inner and stationary cylinder to rotate counterclockwise by applying torque on it, then $\hat{T}$ in Equation 3.17 must be positive. With Equations 3.15 and 3.16, this leads to $T_{r\theta} \geq 0 \; \forall \; r \in [R_\text{i}, R_\text{s}]$. With $\eta > 0$ and $T_{r\theta} \geq 0$ then from Equation 3.18, it is apparent that $(dv_\theta/dr - v_\theta/r) \geq 0$, meaning that it is possible to dismiss the absolute sign in Equation 3.10:

$$\dot{\gamma} = \left| \frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r} \right| = \frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r} \quad \forall \; r \in [R_\text{i}, R_\text{s}] \tag{3.19}$$

Assuming that the Bingham model applies for the test material: $\eta(\dot{\gamma}) = \mu + \tau_\text{o}/\dot{\gamma}$ (see Equation 3.3), then from Equation 3.18 the differential equation to be solved is produced and given by Equation 3.20.

$$\frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r} + \frac{\tau_\text{o}}{\mu} = \frac{\hat{T}}{2\,\pi\,\mu\, r^2\, h} \quad \forall \; r \in [R_\text{i}, R_\text{s}] \tag{3.20}$$

Now, using the above result one can rewrite Equation 3.19 as follows:

$$\dot{\gamma} = \frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r} = \frac{\hat{T}}{2\,\pi\,\mu\, r^2\, h} - \frac{\tau_\text{o}}{\mu} \quad \forall \; r \in [R_\text{i}, R_\text{s}] \tag{3.21}$$

3.3.4 Analytical Results

Because of the *no-slip condition*[9] at the boundary between the solid and the test material in a viscoplastic state, the boundary condition for the velocity is $v_\theta(R_\text{i}) = 0$

[9] **no-slip condition:** The velocity of a fluid at the solid surface is equal to the velocity of the surface; i.e. the fluid sticks to the surface and does not slip relative to it [36]. As demonstrated in Figure 3.5, the inner and outer cylinder consists of ribs aligned as a cylinder. In this way, the test material will form the cylinders. This leads to a larger cohesion (or stickiness) between the two cylinders and the test material in-between.

at the inner cylinder, and $v_\theta(R_\mathrm{s}) = R_\mathrm{s}\omega_\mathrm{o}$ at the outer solid boundary. In other words, it is the Dirichlet boundary condition that is used when solving Equation 3.20:

$$v_\theta(R_\mathrm{i}) = 0 \quad \wedge \quad v_\theta(R_\mathrm{s}) = R_\mathrm{s}\omega_\mathrm{o} \tag{3.22}$$

By dividing Equation 3.20 by r, it becomes $d(v_\theta/r)/dr = \hat{T}/(2\,\pi\,\mu\,r^3\,h) - \tau_\mathrm{o}/(\mu\,r)$. By integration then one gets $v_\theta/r = -\hat{T}/(4\,\pi\,\mu\,r^2\,h) - (\tau_\mathrm{o}/\mu)\ln(r) + C$, where C is the constant of integration. Thereafter, using the condition $v_\theta(R_\mathrm{i}) = 0$ to solve for C produces the following:

$$v_\theta(r) = \frac{\hat{T}\,r}{4\,\pi\,\mu\,h}\left(\frac{1}{R_\mathrm{i}^2} - \frac{1}{r^2}\right) - \frac{\tau_\mathrm{o}\,r}{\mu}\,\ln\!\left(\frac{r}{R_\mathrm{i}}\right) \quad \forall \;\; r \in [R_\mathrm{i}, R_\mathrm{s}] \tag{3.23}$$

Using the condition $v_\theta(R_\mathrm{s}) = R_\mathrm{s}\omega_\mathrm{o}$ in the above, and solving for the torque $\hat{T}$ gives:

$$\hat{T} = \frac{4\,\pi\,\mu\,h}{1/R_\mathrm{i}^2 - 1/R_\mathrm{s}^2}\,\omega_\mathrm{o} + \frac{4\,\pi\,\tau_\mathrm{o}\,h}{1/R_\mathrm{i}^2 - 1/R_\mathrm{s}^2}\,\ln\!\left(\frac{R_\mathrm{s}}{R_\mathrm{i}}\right) \tag{3.24}$$

Because of the boundary condition used, the above two equations are only valid in the domain $r \in [R_\mathrm{i}, R_\mathrm{s}]$. However, the result from the latter equation can be used in the domain $r \in [R_\mathrm{s}, R_\mathrm{o}]$ due to Equation 3.15. Equations 3.23 and 3.24 includes the possibility that the plug is propagating towards the inner cylinder as the angular velocity ω_o is reducing (see Figure 3.10). If no such event is occurring, the condition $R_\mathrm{s} = R_\mathrm{o} =$ constant applies, resulting in the equation shown below.

$$\hat{T} = \frac{4\,\pi\,\mu\,h}{1/R_\mathrm{i}^2 - 1/R_\mathrm{o}^2}\,\omega_\mathrm{o} + \frac{4\,\pi\,\tau_\mathrm{o}\,h}{1/R_\mathrm{i}^2 - 1/R_\mathrm{o}^2}\,\ln\!\left(\frac{R_\mathrm{o}}{R_\mathrm{i}}\right) = H\,\omega_\mathrm{o} + G \tag{3.25}$$

By solving for ω_o in the above, the well known **Reiner-Riwlin**[10] equation [15, 100] is produced. By plotting the measured torque $\hat{T}$ as a function of angular velocity ω_o, one can connect these measured values with a straight line: $\hat{T} = H\,\omega_\mathrm{o} + G$. From its slope H and its point of intersection with the ordinate G, one can calculate the plastic viscosity μ and the yield value τ_o. By algebraic maneuver of Equation 3.25, the plastic viscosity is calculated according to:

$$\mu = \frac{H\,(1/R_\mathrm{i}^2 - 1/R_\mathrm{o}^2)}{4\,\pi\,h} \quad \vee \quad \mu = \frac{H\,(1/R_\mathrm{i}^2 - 1/R_\mathrm{o}^2)}{8\,\pi^2\,h} \tag{3.26}$$

The latter equation of the above, is to be used if **rotational frequency** f_o is used ($\Rightarrow \hat{T} = H\,f_\mathrm{o} + G$) rather than **angular velocity** ω_o ($\Rightarrow \hat{T} = H\,\omega_\mathrm{o} + G$). The relationship is simply $\omega_\mathrm{o} = 2\,\pi\,f_\mathrm{o}$. In similar fashion, the yield value is calculated as shown below.

$$\tau_\mathrm{o} = \frac{G\,(1/R_\mathrm{i}^2 - 1/R_\mathrm{o}^2)}{4\,\pi\,h\,\ln(R_\mathrm{o}/R_\mathrm{i})} \tag{3.27}$$

Combining Equations 3.21 and 3.24 gives the equation for the shear rate:

$$\dot{\gamma} = \frac{2}{r^2}\left(\frac{1}{R_\mathrm{i}^2} - \frac{1}{R_\mathrm{s}^2}\right)^{-1}\left[\omega_\mathrm{o} + \frac{\tau_\mathrm{o}}{\mu}\,\ln\!\left(\frac{R_\mathrm{s}}{R_\mathrm{i}}\right)\right] - \frac{\tau_\mathrm{o}}{\mu} \quad \forall \;\; r \in [R_\mathrm{i}, R_\mathrm{s}] \tag{3.28}$$

[10]The **Reiner-Riwlin** equation is sometimes confused with the **Reiner-Rivlin** equation. The latter is a constitutive equation, which is on the form $\mathbf{T} = -f_1(\mathrm{II}_{\dot{\boldsymbol{\varepsilon}}}, \mathrm{III}_{\dot{\boldsymbol{\varepsilon}}})\,\dot{\boldsymbol{\varepsilon}} - f_2(\mathrm{II}_{\dot{\boldsymbol{\varepsilon}}}, \mathrm{III}_{\dot{\boldsymbol{\varepsilon}}})\,\dot{\boldsymbol{\varepsilon}}\cdot\dot{\boldsymbol{\varepsilon}}$ (see for example textbook by Bird et al. [15]). The terms $\mathrm{II}_{\dot{\boldsymbol{\varepsilon}}}$ and $\mathrm{III}_{\dot{\boldsymbol{\varepsilon}}}$ are the second and the third invariant of the strain rate tensor $\dot{\boldsymbol{\varepsilon}}$.

From Equation 3.28, it is clear that the shear rate is not a constant within the test sample. It is known that suspended particles have the tendency to migrate from region of a high shear rate to the region of low shear rate [9]. As such, the gradient of the shear rate $\nabla\dot{\gamma} = d\dot{\gamma}/dr\,\mathbf{i}_\mathrm{r}$ within the test sample, plays an important role when discussing the possibility of particle migration (see Chapter 10).

$$\frac{d\dot{\gamma}}{dr} = -\frac{4}{r^3}\left(\frac{1}{R_\mathrm{i}^2} - \frac{1}{R_\mathrm{s}^2}\right)^{-1}\left[\omega_\mathrm{o} + \frac{\tau_\mathrm{o}}{\mu}\ln\left(\frac{R_\mathrm{s}}{R_\mathrm{i}}\right)\right] \quad \forall\ r \in [R_\mathrm{i}, R_\mathrm{s}] \tag{3.29}$$

With increasing ω_o, the difference in shear rate $\nabla\dot{\gamma} = d\dot{\gamma}/dr\,\mathbf{i}_\mathrm{r}$ within the test sample will increase, which results in a larger likelihood of particle migration.

3.4 Data Processing

After explaining how the values H and G are converted to the material properties μ and τ_o, it is now natural to clarify how these measured data H and G are actually produced. Depending on the viscometer and on the type of test material, different approaches are applied in this thesis.

3.4.1 Mortar and Cement Paste in ConTec Viscometer 4

Figure 3.6 demonstrates the measured torque $\hat{T}$ as a function of time in the ConTec Viscometer 4. The left illustration demonstrates a mortar measurement (Chapter 6), while the right illustration demonstrates a thixotropic measurement on cement paste (the issue of thixotropy and thixotropic measurements are dealt with in Chapter 9). Figure 3.7, demonstrates further data processing for the mortar case.

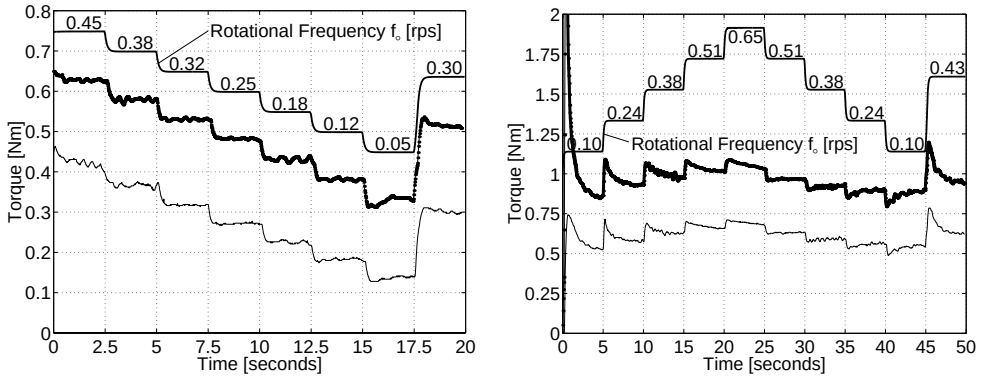


Figure 3.6: Measured torque from the ConTec Viscometer 4. To the left: Mortar with $D_\mathrm{max} = 2\,\mathrm{mm}$. To the right: Thixotropic measurement on cement paste ($D_\mathrm{max} \approx 100\,\mu\mathrm{m}$).

As shown in the left illustration of Figure 3.6, the rotational frequency f_o is decreased in steps, while torque is being logged as a function of time. This approach is designated as *stepwise decreasing shear rate sequence* [67]. The intention of this approach, is to filter out thixotropic effects of the test sample so fast as possible [67].

As shown with Figure 3.7, each discrete rotational frequency f_o has its own duration, designated as *total time*. During this time interval, a number of 150 distinct torque points are logged. This time duration is further divided in two parts [34], the

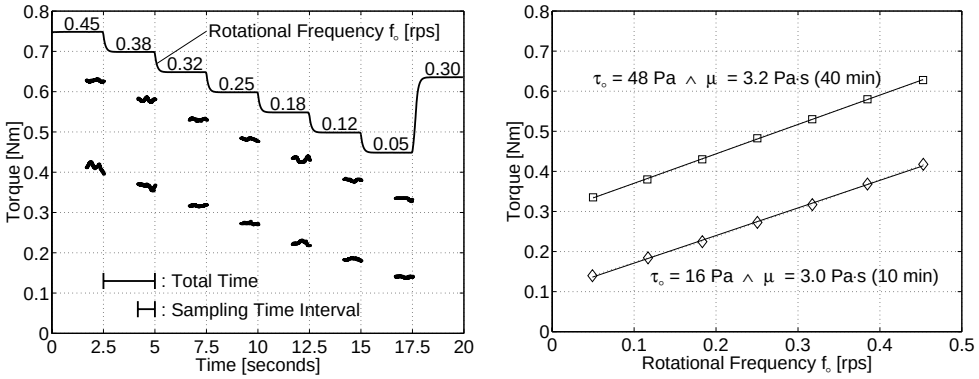


Figure 3.7: To the left: Logged torque at the sampling time interval. The whole series is shown in Figure 3.6. To the right: After processing the data, the torque $\hat{T}$ as a function of rotational frequency f_o, is generated: $\hat{T} = H\, f_o + G$.

transient time interval and the *sampling time interval*, which of the latter consist of 50 torque points. The **average of these 50 points** are then used in forming a single torque point $\hat{T}$. A number of $2 \cdot 7 = 14$ such points are shown in the right illustration of Figure 3.7. The reason for using only the last 50 logged points of the total 150, is to give the test sample time to reach equilibrium state[11], i.e. time independence in the velocity function v_θ, prior to data processing. Now, with these $2 \cdot 7 = 14$ torque points, it is possible to construct two functions of the type $\hat{T} = H\, f_o + G$, through a linear regression. For example, at "10 min" the values produced are $H = 0.688\,\mathrm{Nm} \cdot \mathrm{s}$ and $G = 0.102\,\mathrm{Nm}$. Using Equations 3.26 and 3.27, with the geometry of $R_i = 8.5\,\mathrm{cm}$, $R_o = 10.1\,\mathrm{cm}$ and $h = 11.6\,\mathrm{cm}$, results in $\mu = 3.03\,\mathrm{Pa} \cdot \mathrm{s}$ and $\tau_o = 16.4\,\mathrm{Pa}$.

3.4.2 Concrete in ConTec BML Viscometer 3

Barrier Restraint of the Inner and Outer Cylinder

When using the ConTec **BML** Viscometer 3, the characteristic thickness of the flow is $D_{\mathrm{flow}} = \Delta R = R_o - R_i = 14.5\,\mathrm{cm} - 10\,\mathrm{cm} = 45\,\mathrm{mm}$. Using fresh concrete with $D_{\max} = 16\,\mathrm{mm}$ as a test sample, means that the condition of $D_{\mathrm{flow}} \approx 3D_{\max}$ is achieved. [The term D_{flow} is the characteristic thickness of the flow, while $D_{\max}$ is the diameter of the largest solid particle]. There is one drawback with the condition of $D_{\mathrm{flow}} \approx 3D_{\max}$: With a barrier restraint of the inner and outer cylinder, combined with the lack of space for the larger aggregate particles, these solid particles will have a larger and a more frequent momentum exchange with each other for the given rotational frequency f_o (i.e. larger $\Delta \mathrm{p}_{\mathrm{dc}}$ and $\tilde{N}_{\mathrm{dc}}$ values; see the next paragraph). This is due to their lack of motional freedom in avoiding such a strong mechanical solid particle interaction; i.e. due to the lack of space that is needed for the larger aggregate particles to move freely around and in such manner, avoid an abnormal high momentum exchange.

In Section 2.4.1, the shear viscosity η for cement paste is generated. This is shown with Equation 2.25 on Page 26. With this equation, it is demonstrated that the momentum exchange $\Delta \mathrm{p}_{\mathrm{dc}}$ by direct collision, between the larger cement particles,

[11] In each and every mortar test, equilibrium was checked by plotting illustrations like what is shown in Figure 3.6. When equilibrium is not present, the corresponding torque point $\hat{T}$ is omitted.

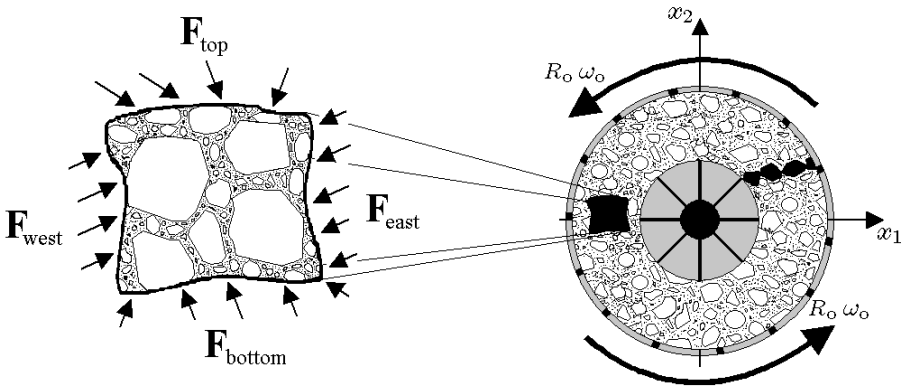


Figure 3.8: A CP of concrete, inside the ConTec **BML** Viscometer 3 (to the left) and formation of a bridge of the larger aggregates inside the same viscometer (to the right).

is one of the physical processes that generates shear viscosity η. Also, the number of such collisions $\tilde{N}_{\text{dc}}$ plays an equally important role. Extrapolate this result to the fresh concrete, its shear viscosity becomes as $\eta \propto \tilde{N}_{\text{dc}}\,\Delta\text{p}_{\text{dc}}$, where $\Delta\text{p}_{\text{dc}}$ is now the momentum exchange between the larger aggregate particles by direct collision. With the above-mentioned barrier restraint of the inner and outer cylinder, a larger and a more frequent momentum exchange will result between the larger aggregate particles (i.e. larger $\Delta\text{p}_{\text{dc}}$ and $\tilde{N}_{\text{dc}}$ values). This results in a larger shear viscosity $\eta \rightarrow \eta + \Delta\eta$ than actually applies for the concrete when flowing more freely inside a mold or formwork.

Additional consequence of the above-mentioned barrier restraint is that a bridge of the larger aggregates forms frequently between the outer and inner cylinder, resulting in direct torque transportation from the outer cylinder to the inner cylinder. This is shown with the right illustration of Figure 3.8. In this case, the momentum exchange between the larger aggregate particles $\Delta\text{p}_{\text{dc}}$ is at maximum, for the given rotational frequency f_o.

It is clear that the barrier restraint of the inner and outer cylinder, gives a larger and more frequent momentum exchange between the larger aggregate particles. This results in a fluctuation in logged torque as shown in Figure 3.9. With this type of increased momentum exchange (that elevates the torque curve $\hat{T} \rightarrow \hat{T}+\Delta\hat{T}$) and using the **average of the 50** distinct logged torque points (as is done for the mortar case), a larger shear viscosity $\eta \rightarrow \eta + \Delta\eta$ will be measured. To reduce the addition $\Delta\eta$, only the **average of the 10 lowest** torque points (of the pre-mentioned 50) is used when forming a single torque point $\hat{T}$. This approach has always been (and is still) provided in the software FreshWin, that is supplied with the ConTec viscometers.

When using the ConTec Viscometer 4, the characteristic thickness of the flow is $D_{\text{flow}} = \Delta R = R_\text{o} - R_\text{i} = 10.1\,\text{cm} - 8.5\,\text{cm} = 16\,\text{mm}$. Using mortar with $D_{\text{max}} = 2\,\text{mm}$ as a test sample, means that the condition $D_{\text{flow}} = 8D_{\text{max}}$ is achieved. Hence, there is a much larger motional freedom present for the individual solid particles in avoiding an abnormal high momentum exchange. This is clear with Figure 3.6, which shows a much smoother torque curve, relative to Figure 3.9. In some literature [17], the condition of $D_{\text{flow}} \geq 10D_{\text{max}}$ is generally recommended for coaxial cylinders viscometers. Relative to the current text, the condition of $D_{\text{flow}} \geq 8D_{\text{max}}$ seems to be sufficient. As such, for mortar, the **average** of the data produced in the sampling time interval (the pre-mentioned 50 logged torque points) can be assumed to represent

the actual response of the test material (when flowing inside a formwork) and is therefore used when forming a single torque point $\hat{T}$, shown in the right illustration of Figure 3.7. For the cement paste in Chapter 9, the condition $D_{\text{flow}} \approx 160 D_{\max}$ is valid, since $D_{\max} \approx 100\,\mu$m. Therefore, a discrete logged torque point is assumed to represent the correct response of the test material.

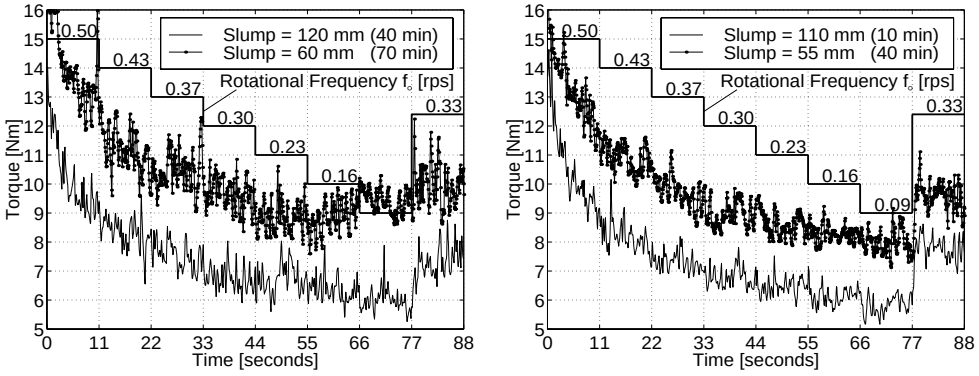


Figure 3.9: A raw measuring data from the ConTec **BML** Viscometer 3 [$w/c = 0.6$; 0.1% sbwc of **VHMW Na**]. The same mix design applies for both illustrations (Table 4.3 on Page 78), however with $D_{\max} = 16\,\text{mm}$ to the left and $D_{\max} = 11\,\text{mm}$ to the right. As is apparent in both illustrations, no "equilibrium state" is present for the concrete results shown here. The continuous decrease in torque is related to gravel migration, as discussed in Chapter 10.

When replacing the $11 - 16\,\text{mm}$ aggregates with the $8 - 11\,\text{mm}$ ones (see Section 4.2.3), the condition $D_{\text{flow}} \approx 3D_{\max}$ is changed to $D_{\text{flow}} \approx 4D_{\max}$. Figure 3.9 demonstrates that the registered fluctuation in torque becomes smaller after such replacement. This is to be expected since with $D_{\text{flow}} \approx 4D_{\max}$, the larger solid particles gain a larger motional freedom in avoiding the abnormal high momentum exchange. The fluctuation in logged torque could be further reduced by increasing the amount of binder and/or by increasing the amount plasticizer. The former would increase the average distance between the larger solid particles, while the latter would result in a better lubrication between these particles.

Gravel Migration

According to Barnes and coworkers [9], migration of the suspended particles frequently occurs, from the region of high shear rate, to the region of low shear rate (i.e. in the direction of $-\nabla\dot{\gamma}$). This phenomenon was always observed after a viscometric measurement on the concrete batches of this thesis (see Figure 10.2, Page 238). In Chapter 10, it is investigated how much this phenomenon influence the test results. It is concluded in Section 10.2.3, that the best way to calculate the viscometric values is first by replacing $R_{\text{o}} = 14.5\,\text{cm}$ with $R_{\text{o}} = 12\,\text{cm}$ in Equations 3.26 and 3.27. Also, when generating the slope H and the point of intersection with the ordinate G, only the last 5 torque points $\hat{T}$ (of total 7) is used. For the particular case in Figure 3.9, this means that selected[12] logged torque values only between 22 and 77 seconds are used. In Chapter 10, the resulting new viscometric values are designated with $\tau_{\text{o}}^{\clubsuit}$ and $\mu^{\clubsuit}$. In all other chapters of this thesis, these values are always designated with τ_{o}

[12] Data from the *sampling time interval* is only used (see Figure 3.7).

and μ. In Chapter 10, four different ways are used when calculating the viscometric values. Three of them are listed below. In these calculations the geometrical values $R_\mathrm{i} = 10\,\mathrm{cm}$ and $h = 19.9\,\mathrm{cm}$ are always used.

1. $\tau_\mathrm{o}^{\bigcirc} \wedge \mu^{\bigcirc}$: In generating H and G, the 5 torque points $\hat{T}$ (of total 7) are used. Using H and G in Equations 3.26 and 3.27 and using $R_\mathrm{o} = 14.5\,\mathrm{cm}$ as the outer cylinder, gives the viscometric values $\tau_\mathrm{o}^{\bigcirc} \wedge \mu^{\bigcirc}$.

2. $\tau_\mathrm{o}^{\clubsuit} \wedge \mu^{\clubsuit}$ ($\approx \tau_\mathrm{o}^{\diamondsuit} \wedge \mu^{\diamondsuit}$): In generating H and G, the 5 torque points $\hat{T}$ (of total 7) are used. Using H and G in Equations 3.26 and 3.27 and using $R_\mathrm{o} = 12\,\mathrm{cm}$ as the outer cylinder, gives the two viscometric values $\tau_\mathrm{o}^{\clubsuit} \wedge \mu^{\clubsuit}$. This approach is used in Chapters 5 and 6.

3. $\tau_\mathrm{o}^{\square} \wedge \mu^{\square}$: In generating H and G, all the 7 torque points $\hat{T}$ (of total 7) are used. Using H and G in Equations 3.26 and 3.27 and using $R_\mathrm{o} = 14.5\,\mathrm{cm}$ as the outer cylinder, gives the two viscometric values $\tau_\mathrm{o}^{\square} \wedge \mu^{\square}$. If the fresh concrete were in a homogeneous state (meaning that no gravel migration is occurring and no segregation of any kind) and without any thixotropic behavior, this approach would give the correct viscometric values.

3.4.3 Parameter Setup

As shown in Table 3.1, for concrete, mortar and cement paste, different parameters are used in controlling for how long time the torque is logged and then processed. Many of the values shown are chosen from the default setup of the viscometers.

The *total time* for concrete, consist of 9 seconds, with *transient time interval* and *sampling time interval* of 3.3 and 5.7 seconds, respectively. However, for the concrete sample in Chapter 10, the *total time* consist of 11 seconds, with *transient time interval* and *sampling time interval* of 3.7 and 7.3 seconds, respectively.

Table 3.1: Parameter setup used in the ConTec viscometers.

	Concrete	Concrete (Chapter 10)	Mortar	Cement Paste
Transient time interval	3.3 s	3.7 s	1.67 s	0 s
Sampling time interval	5.7 s	7.3 s	0.83 s	5 s
Number of logged points	50	50 (150)	50 (150)	150
Data processing	10 lowest	10 lowest	average	thixotropic
Total time	9 s	11 s	2.5 s	5 s
Pre-rotation	≈ 10 s	≈ 10 s	0 s	0 s
f_max	0.50 rps	0.50 rps	0.45 rps	0.65 rps
f_min	0.09 rps	0.09 rps	0.05 rps	0.10 rps
Number of $\hat{T}$ points	5 (7)	5 and 7	7	continuous

With the parameter setup used for the ConTec **BML** Viscometer 3, the outer cylinder always rotates for about 10 seconds at 0.5 rps prior to any data logging. This is shown in Table 3.1 with the row "Pre-rotation". However, this type of pre-rotation is never applied for the ConTec Viscometer 4. The terms f_max and f_min describes the maximum and minimum rotational frequency applied to the outer cylinder. For concrete and mortar, these values are determined in Section 4.4.1. The "Number of

$\hat{T}$ points" describes how many torque points are generally used in extracting the G and H values. For example in Figure 3.7, the number is 7 for mortar. For cement paste, this field is marked as "continuous", meaning that all available logged torque points ($10 \cdot 150 = 1500$) is used when analyzing the thixotropic properties of the test material (see for example Figure 9.3 on Page 214).

3.5 Influence of Plug State

3.5.1 Introduction

The background of this section is related to the concern when some of the mortar mixes in this thesis, consist of rather high ratio of yield value to plastic viscosity τ_o/μ. Such material is prone to form a plug during the viscometric measurement. With a plug, it is meant a solid state as defined in Section 3.2. The material in the solid state rotates as a rigid body, inside the viscometer ($v_\theta = r\,\omega_o$).

Eliminating the plug state simply by increasing the angular velocity ω_o, is not considered to be a good idea, because according to Equation 3.29, such action would increase the difference in the shear rate $\nabla\dot{\gamma} = d\dot{\gamma}/dr\,\mathbf{i}_r$ within the test sample. With such increase, migration of the larger sand particles would be more likely, from the region of high shear rate to the region of low shear rate (see the discussion in Section 10.1). In some very rear occasions, particle migration occurred for the mortars of this thesis. This occurred only with a very large τ_o/μ-ratio. As shown with Equation 3.29, increased τ_o/μ-ratio results in a larger difference in the shear rate $\nabla\dot{\gamma} = d\dot{\gamma}/dr\,\mathbf{i}_r$. The left illustration of Figure 10.2 (Page 238) shows an example of such particle migration for mortar, using the standard rotational frequency f_o shown for example in the left illustration of Figure 3.6.

The effect of plug in concrete measurements is not considered, because the prementioned gravel migration is dominating in introducing an error to the viscometric values τ_o and μ (see the center illustration of Figure 10.2).

3.5.2 Classification of Plug

With plug state it is meant that some part of the test material is in a solid state. This is shown with Figure 3.10, where the plug state exist in the domain $r \in [R_s, R_o]$. With Equation 3.28, it is apparent that the shear rate $\dot{\gamma}$ decreases with increasing radii r. Therefore, when applying the *stepwise decreasing shear rate sequence* (Section 3.4.1), the condition $\dot{\gamma} = 0$ will begin at the outer cylinder. According to Equation 3.5, this means that the plug will begin at the outer cylinder and propagate towards the inner cylinder as the angular velocity ω_o is further decreased ($\omega_o = 2\,\pi\,f_o$). Designating the location of the boundary between a viscoplastic and a solid state with $r = R_s$ and writing $\dot{\gamma}|_{r=R_s} = 0$ in Equation 3.28, gives Equation 3.30 after some simple algebraic maneuvers.

$$\omega_o = \frac{\tau_o}{\mu}\left[\frac{1}{2}\left(\frac{R_s^2}{R_i^2} - 1\right) - \ln\left(\frac{R_s}{R_i}\right)\right] \tag{3.30}$$

By creating a vector that represents the potential plug radii $\mathbf{R}_s = [R_i, \ldots R_o]$ and putting it in Equation 3.30, one creates a corresponding vector of angular velocity $\boldsymbol{\omega_o}(\mathbf{R}_s) = [\omega_o|_{R_s=R_i}, \ldots \omega_o|_{R_s=R_o}]$ where plug state is active. From this it is apparent that each element in the vectors $\mathbf{R}_s$ and $\boldsymbol{\omega_o}$ corresponds to each other. The last

element in the vector $\boldsymbol{\omega_o}(\mathbf{R}_s)$ represents the angular velocity when plug state starts to form at the outer cylinder $r = R_o$ and is designated with ω_o^p. In other words, the definition $\omega_o^p \equiv \omega_o|_{R_s=R_o}$ applies relative to Equation 3.30. Putting the two vectors $\mathbf{R}_s$ and $\boldsymbol{\omega_o}(\mathbf{R}_s)$ in Equation 3.24, produces the corresponding torque vector $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega_o})$. Plotting $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega_o})$ as a function of $\boldsymbol{\omega_o}(\mathbf{R}_s)$ produces the torque profile when the plug state is extending from the outer cylinder R_o towards the inner cylinder R_i. An example of such torque profile is shown in Figure 3.10.

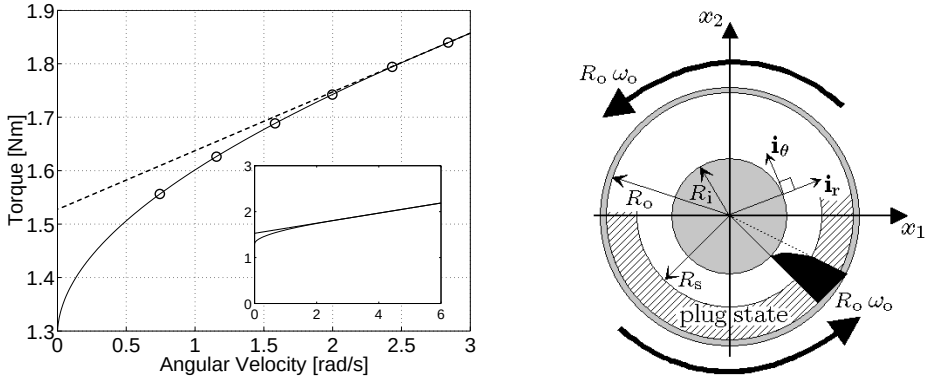


Figure 3.10: Plot of $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega_o})$ as a function of $\boldsymbol{\omega_o}(\mathbf{R}_s)$. The viscometric values are $\tau_o = 245.3\,\text{Pa}$ and $\mu = 3.05\,\text{Pa}\cdot\text{s}$ and as such, then $\omega_o^p = 2.7\,\text{rad/s}$. The geometry consist of $(R_i, R_o, h) = (8.5, 10.1, 11.6)\,\text{cm}$.

The dashed line in Figure 3.10 is an extrapolation from the torque when no plug is occurring, which is readily seen with the small incorporated figure. This line is generated by using Equation 3.25. Each circle ○ represent a discrete torque point, as would be measured when the plugged zone is extending towards the inner cylinder.

If a result like the solid line shown in Figure 3.10 is available, it is possible to calculate the yield value from the new point of intersection alone. Putting Equation 3.30 in Equation 3.24 and then taking the limit "$\lim_{R_s \to R_i} \hat{T} = G_o$", where G_o is the new point of intersection, gives the following: $\tau_o = G_o/(2\,\pi\,R_i^2\,h)$. By also using Equation 3.27, the relationship $G/G_o = 2\,\ln(R_o/R_i)/(1 - R_i^2/R_o^2)$ is produced.

The torque $\hat{T}(\omega_o^p)$ in which plug first occurs can be calculated by combining Equations 3.30 and 3.24, and using $R_s = R_o$. The result is $\hat{T}(\omega_o^p) = 2\,\pi\,R_o^2\,h\,\tau_o$. Hence the relationship between $\hat{T}(\omega_o^p)$ and G_o is given by $\hat{T}(\omega_o^p)/G_o = R_o^2/R_i^2$. For the example in Figure 3.10, then $G_o = 1.29\,\text{Nm}$, $G = 1.53\,\text{Nm}$ and $\hat{T}(\omega_o^p) = 1.82\,\text{Nm}$.

For the given angular velocity ω_o, the location of the solid boundary R_s can be calculated by applying the Newton-Raphson iteration algorithm [21] on Equation 3.30, or more precisely on Equation 3.31, shown below.

$$\tilde{f}(R_s) = \frac{\tau_o}{\mu}\left[\frac{1}{2}\left(\frac{R_s^2}{R_i^2} - 1\right) - \ln\left(\frac{R_s}{R_i}\right)\right] - \omega_o = 0 \tag{3.31}$$

Basically this algorithm consist of iterating Equation 3.32, until $|R_s^{n+1} - R_s^n|$ is less than some specific value (n is the iteration index). The first guess usually consists of either the inner $R_s^0 = R_i$ or the outer $R_s^0 = R_o$ radius.

$$R_s^{n+1} = R_s^n - \frac{\tilde{f}(R_s^n)}{d\tilde{f}(R_s^n)/dR_s} \tag{3.32}$$

Equations 3.31 and 3.32 are frequently used in Chapter 8, when verifying the numerical result produced by the software VISCOMETRIC-VISCOPLASTIC-FLOW. These two equations are also used in Chapter 10.

3.5.3 Error Generated by Plug Flow

Figure 3.11 shows measured torque $\hat{T}$ as a function of angular velocity ω_o for the same mortar specimen at four different times, using the CONTEC VISCOMETER 4. The test material is a $0-2$ mm mortar with lignosulfonate [**LMW Na**; 0.6% sbwc; $T_o = 38°C$; $w/c = 0.4$; OPC]. The measurements are made at 10, 40, 70 and 100 minutes after water addition. The measured torque $\hat{T}$ increases with these four time points because of the chemical reactions of the cement clinker and water (see Section 2.5.1).

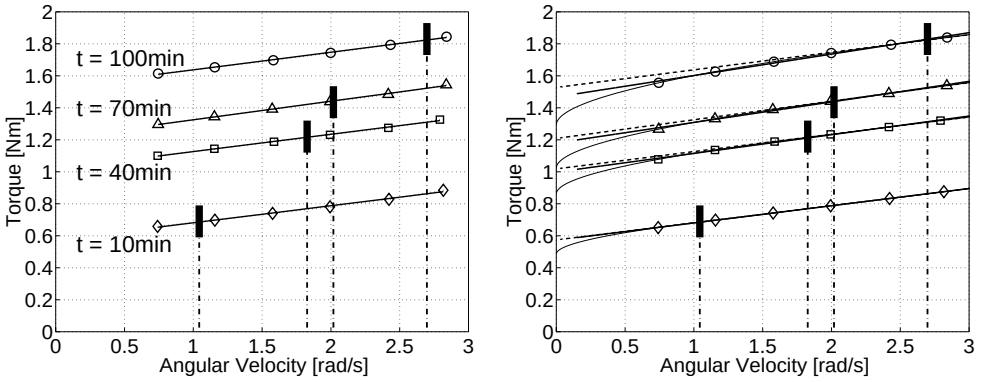


Figure 3.11: To the left: Measured torque on the same mortar sample. The thick vertical lines pinpoint the angular velocity when plug state starts to form (i.e. ω_o^p), assuming that the two viscometric values τ_o and μ are correct. To the right: Plot of $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega}_\mathbf{o})$ as a function of $\boldsymbol{\omega}_\mathbf{o}(\mathbf{R}_s)$. Both illustrations are from [143].

From the slope H and the point of intersection with the ordinate G of the lines in Figure 3.11, one can calculate the plastic viscosity μ and the yield value τ_o with Equations 3.26 and 3.27 ($(R_i, R_o, h) = (8.5, 10.1, 11.6)$ cm). The resulting viscometric values are shown in Table 3.2. The term R^2 shown there, is the coefficient of determination [148] (R is referred as the correlation coefficient [148]). Also shown is the term ω_o^p, which is the angular velocity when plug state starts to form at the outer cylinder. This ω_o^p-calculation is based on the assumption that the viscometric values τ_o and μ are correct to begin with. The location of the ω_o^p values are marked with a thick vertical lines in Figure 3.11.

Table 3.2: Measured viscometric values.

Minutes	10	40	70	100
τ_o [Pa]	92.2	163.5	194.1	245.3
μ [Pa · s]	2.96	3.00	3.22	3.05
R^2	0.994	0.998	0.998	0.998
ω_o^p[rad/s]	1.04	1.83	2.02	2.70

Using the values in Table 3.2 when plotting $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega}_\mathbf{o})$ as a function of $\boldsymbol{\omega}_\mathbf{o}(\mathbf{R}_s)$, in the same manner as demonstrated with Figure 3.10, provides the result shown in

the right illustration of Figure 3.11. The markers represent the discrete torque points as they should be measured if the τ_o and μ values in Table 3.2 where correct (the dashed line is the same as the solid line shown in the left illustration). By comparing the markers in the left and the right illustration of Figure 3.11, it is clear that they do not always overlap each other. Therefore, it is evident that some of the τ_o and μ values in Table 3.2 are incorrect to begin with. This applies especially for the test conducted at 100 minutes after water addition. The solid straight lines in the right illustration, are linear regression of the markers. Putting the slopes H and the point of intersections G of these solid lines, into Equations 3.26 and 3.27, produces a new set of yield value $\tau_o^{(2)}$ and plastic viscosity $\mu^{(2)}$, shown in Table 3.3.

Table 3.3: Recalculated viscometric values.

Minutes	10	40	70	100
$\tau_o^{(2)}$ [Pa]	91.9	160.3	189.5	235.7
Err_τ [%]	0.3	2	2.4	3.9
$\mu^{(2)}$ [Pa · s]	2.98	3.24	3.56	3.71
Err_μ [%]	0.7	8	10.4	21.9
R^2	1.000	0.998	0.996	0.994

Since the correct values of the tested sample are unknown to begin with, it is difficult to calculate exact magnitude of error to accompany the values in Table 3.2. However by comparing the values in Tables 3.2 and 3.3, one could gain an estimation of the error, at least in the order of magnitude. The term "Err" shown in this table represents the percentage difference between the values in both tables (see Equation 3.33).

$$\mathrm{Err}_\tau = (|\tau_o - \tau_o^{(2)}|/\tau_o) \cdot 100\% \quad \wedge \quad \mathrm{Err}_\mu = (|\mu - \mu^{(2)}|/\mu) \cdot 100\% \tag{3.33}$$

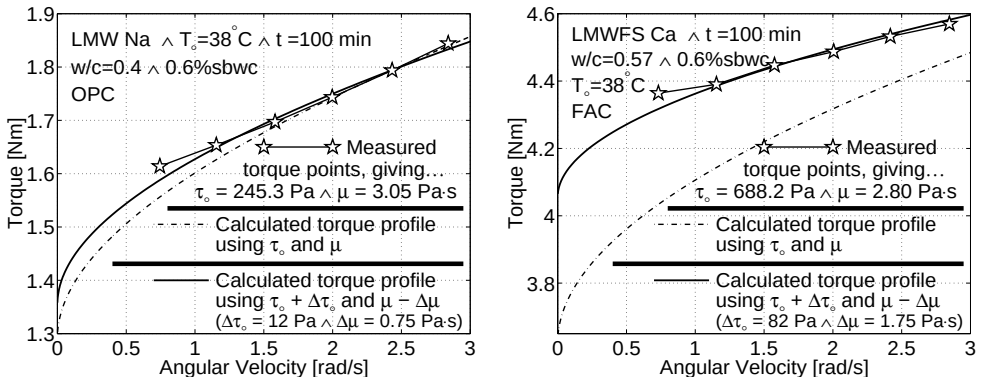


Figure 3.12: By changing the viscometric values from $(\tau_o \wedge \mu)$ to $(\tau_o + \Delta\tau_o \wedge \mu - \Delta\mu)$, the torque profile $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega}_o)$ can overlap the measured torque points. The viscometric values τ_o and μ are calculated by Equations 3.26 and 3.27. These two equations assumes that no plug is occurring inside the test material.

There exists a time consuming method in extracting a more correct viscometric values τ_o and μ from the tested material. It consist of manually changing the τ_o and

μ values until the torque profile $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega}_\mathbf{o})$ overlaps the measured torque points. This is done in Figure 3.12. The left illustration is the same experimental case, as the "$t = 100\,\text{min}$"-case in Figure 3.11. For this particular case, the initial yield value has to be increased by 4.9% and the plastic viscosity decreased by 24.6% to make the torque profile $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega}_\mathbf{o})$ overlap most of the measured torque points. These two percentage values are approximately the same as shown in Table 3.3, indicating that the error estimation made in this table is not so bad. The initial τ_o/μ-ratio for this case, is about $245.3\,\text{Pa}/3.05\,\text{Pa}\cdot\text{s} = 80\,\text{s}^{-1}$. As will be apparent in Section 6.4, the condition $\tau_o/\mu \leq 100\,\text{s}^{-1}$ will apply in most cases for mortar. Hence, most often, it is not to be expected that the plug flow will generate any error of dramatic magnitude. However, in a few occasions, the τ_o/μ-ratio will become as high as around $300\,\text{s}^{-1}$. The right illustration of Figure 3.12 demonstrates such incidence. For this particular case, the initial yield value has to be increased by 11.9% and the plastic viscosity decreased by 62.5% to make the torque profile $\hat{\mathbf{T}}(\mathbf{R}_s, \boldsymbol{\omega}_\mathbf{o})$ overlap most of the measured torque points. Fortunately, such a high τ_o/μ-ratio applies rarely and as such, plug flow is not considered to produce too much of a problem.

Discussion and Conclusion

Table 3.4 shows the viscometric values from Table 3.2 with the errors from Table 3.3. These errors are presented as an order of magnitude rather than of the actual (and unknown) values. From Table 3.4, it is clear that the plastic viscosity is more sensitive to the error produced by plug, relative to the yield value. When plug is occurring and the viscometric values are extracted by Equations 3.26 and 3.27, the resulting τ_o value becomes lower (by about 1%) than the true yield value of the tested material. Also, the resulting μ value becomes larger (by about 10%) than the true plastic viscosity of the tested material (see also the left illustration of Figure 3.12). This is because of how Equations 3.26 and 3.27 do not include the possibility of plug flow. Since the viscometric values shown in Table 3.4, are quite typical in this work, it can be concluded that even if plug is occurring in some of the measurements, it does not introduce a very large error to the viscometric values τ_o and μ.

In some extreme cases, a yield value of around $\tau_o \approx 700\,\text{Pa}$ and plastic viscosity of around $\mu \approx 3\,\text{Pa}\cdot\text{s}$ is measured. In such cases, the order of magnitudes in error produced, are 10% and 100%, respectively (see the right illustration of Figure 3.12).

Table 3.4: Measured viscometric values.

Minutes	10	40	70	100
τ_o [Pa]	92	164	194	245
Error [%]	10^{-1}	1	1	1
μ [Pa · s]	3.0	3.0	3.2	3.1
Error [%]	1	10	10	10

A discussion about the errors made by plug flow and by air entrainment (Section 5.5) is made in Section 6.4, before considering the overall experimental results of mortars.

Chapter 4

Experimental Program

4.1 Introduction

The batches of this thesis are basically made of four constituents, namely cement, water, aggregates and lignosulfonates (however, in Chapter 9, the aggregates are not included). When mixed together, these constituents form the particle suspension of interest. This is demonstrated with Figure 4.1.

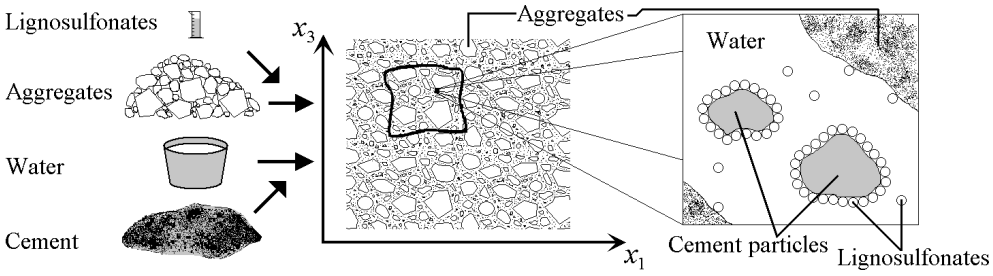


Figure 4.1: When mixing the different constituents, namely the water, cement, lignosulfonate and aggregates, the particles suspension is formed (see also Figure 2.2, Page 12).

Four issues are dealt with in Chapter 4. First, a description about each constituent is made, which are used in the batches. Secondly, the mix design of the different batch type is described. With mix design, it is meant the relative proportions of each constituent used in the different type of batch. Thirdly, the test methods used, in classifying the rheological behavior of the batches is described. Finally, the overall test program of this thesis is introduced. The purpose of the test program is presented in Section 1.2.1 and the result of these tests are shown in Chapters 5 and 6.

4.2 Constituents

4.2.1 Cement Types (OPC, FAC)

Two different cement types are used in this project and are both produced by Norcem AS, which is part of the German cement and building materials group Heidelberger Zement AG. The two cement types are the Norcem Standard Cement (OPC)

and the NORCEM FLY ASH CEMENT (FAC). These two cements are one of the most common types used in Norway.

The NORCEM STANDARD CEMENT (EN 197-1 CEM I 42.5R) is Ordinary Portland Cement. As such, this cement type will be often designated as OPC in this thesis. The loss of ignition for this cement is 2.4%. About 5% of the OPC consist of gypsum, i.e. of $C\bar{S}H_2$. The density and fineness (Blaine) of the OPC is $\rho_{opc} = 3120\,kg/m^3$ and $340\,m^2/kg$, respectively. Table 4.1 demonstrates the magnitude of the different clinker constituents used in the NORCEM STANDARD CEMENT. A description about the clinker minerals is made in Section 2.5.1 (see also Footnote 24, Page 34 about the clinker minerals C_3S, (β-) C_2S, C_3A and C_4AF).

Table 4.1: Clinker constituents used in the OPC.

C_3S	C_2S	C_3A	C_4AF	Na_2O-eqv
60%	15%	7.5%	10%	0.95%

The NORCEM FLY ASH CEMENT (EN 197-1 CEM II A/V 42.5R) consist of the same clinker type as used in the NORCEM STANDARD CEMENT (see Table 4.1). This cement type will be frequently designated as FAC in this thesis. The FAC consist of about 80% OPC and of 20% fly ash (imported from Denmark). The density and fineness (Blaine) of the FAC is $\rho_{fac} = 2950\,kg/m^3$ and $437\,m^2/kg$, respectively. The loss of ignition is 1.3%.

4.2.2 Lignosulfonates

The lignosulfonates used in this thesis are produced by BORREGAARD LIGNOTECH, Norway (a member of the ORKLA group). Six types of lignosulfonates (LS) are used in this work, and the right illustration of Figure 4.2 gives a schematic illustration of how the different polymer types originate from the same base lignosulfonate product **LMWFS Ca**. The following designations will be used for the different lignosulfonate products:

- **VHMW Na** ⇒ **V**ery **H**igh **M**olecular **W**eight **Na**-Lignosulfonate
- **HMW Na** ⇒ **H**igh **M**olecular **W**eight **Na**-Lignosulfonate
- **HMW Ca** ⇒ **H**igh **M**olecular **W**eight **Ca**-Lignosulfonate
- **LMW Na** ⇒ **L**ow **M**olecular **W**eight **Na**-Lignosulfonate
- **LMW Ca** ⇒ **L**ow **M**olecular **W**eight **Ca**-Lignosulfonate
- **LMWFS Ca** ⇒ **L**ow **M**olecular **W**eight **F**ull **S**ugar **Ca**-Lignosulfonate

Within the recomendend dosage by BORREGAARD LIGNOTECH, then according to ASTM C494, both the **HMW Na** and **HMW Ca** are designated as retarding high-range water-reducer (superplasticizer) of type G. No ASTM designation is assigned to the **VHMW Na** lignosulfonate, since it is an experimental product, but the performance is typical of type F. According to ASTM C494, the **LMW Na** and **LMW Ca** are designated as retarding water-reducer of type A. The **LMWFS Ca** is also an ASTM C494 type A performing lignosulfonate.

Description about the chemical and physical properties of these six lignosulfonate types will now follow.

General Properties

The raw material for production of the lignosulfonate samples is spruce, which is a softwood [18]. The wood cell of spruce (and all other tree types) consist of mainly cellulose, hemicellulose and lignin. A simplified picture describes a skeleton of cellulose surrounded by other substances functioning as matrix (hemicellulose) and encrusting material (lignin). The word lignin is derived from the Latin word lignum, meaning wood. The important role of lignin is to acts as an encrusting or a binder for the wood cell, giving the tree its structural strength. In addition, it performs other functions which are essential [107].

At the Borregaard plant in Norway, the spruce wood is processed to liberate the cellulose fibers. The latter is further processed to give cellulose of different quality and properties. The cellulose is liberated from the lignin and from the hemicellulose by a chemical pulping process. This chemical process consist of mixing a solution of calcium-hydrogen-sulphite ($CaHSO_3^-$) and water, with grind spruce wood [19]. Under the condition of 140°C and pH $\approx$ 1 to 2, the cellulose is effectively liberated. After the liberation process, the cellulose is separated from the rest of the solution for further treatment. The rest solution, which is called the lignosulfonate intermediate, contains sulfonated lignin with a wide range of molecular size, sugars[1] (mainly pentoses and hexoses), inorganic salts and small amounts of other chemicals from the spruce wood. Since the calcium (Ca) is the base in the above-mentioned chemical pulping process, all salts produced are calcium salts [19]. This means that the base lignosulfonate product, in the lignosulfonate intermediate, is a calcium-lignosulfonate. This lignosulfonate intermediate is designated with **LMWFS Ca**. As shown in the right illustration of Figure 4.2, three different processes are applied separately or in combination to achieve or improve the special properties of **LMWFS Ca** [19].

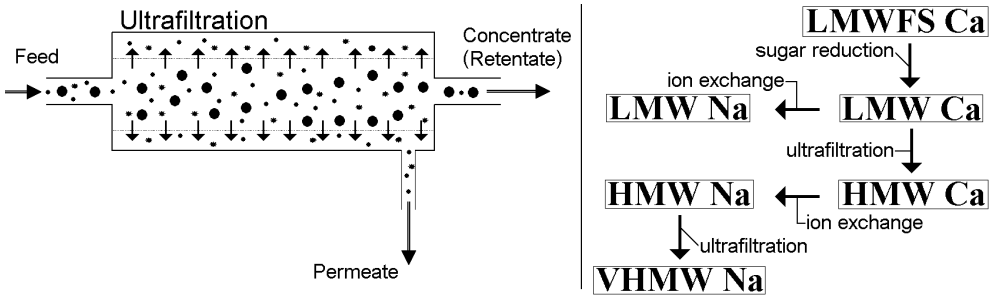


Figure 4.2: To the left: A schematic illustration of ultrafiltration. To the right: All lignosulfonates used in this thesis, originates from the same base product, namely the **LMWFS Ca**.

The first process shown in Figure 4.2, is known as **fermentation** (or sugar reduction). There the hexose is converted into alcohol by fermentation and the alcohol is then removed by distillation. Some amount of remaining sugars are converted into sugar acid, which have a less retarding effect on cement hydration, than of sugar. Since the **LMWFS Ca** product does not go under the fermentation process (i.e. sugar reduction), its sugar content is high. This is indicated with the mark "FS", meaning Full Sugar.

The second process is the so-called **ion exchange**, which consist of adding sodium sulfate into a solution of (fermented) calcium-lignosulfonate and water. The calcium

[1]The sugars have a retarding effect on the freshly mixed concrete (see Page 128).

ion detach itself from the lignosulfonate core while the sodium ion takes its place instead. Thereafter, calcium sulfate precipitate and is removed. Sugar content is normally slightly reduced during this process and also the content of insoluble salts.

The third process is known as **ultrafiltration**. This method consist of molecular filtration through a very fine membrane as shown in the left illustration of Figure 4.2. During this process, large amount of sugars, monomers and smaller sized polymers are removed. Also, the resulting high molecular weight product (the concentrate) will have a higher lignin content than the feed.

Table 4.2, demonstrates the main characteristics of the lignosulfonates. The term DOS is a designation for **Degree of Sulfonation**. It presents the utilization of available chemical groups (present in the lignosulfonate) in attaching the (active) anion SO_3^- to it. The general trend is that as the molecular weight increases, then larger proportion of these chemical groups are inside the bulk of the lignosulfonate and hence are not available in attaching the (active) anion to it [18]. In this sense, the DOS value describes the ratio of surface area to volume of the lignosulfonate [84]. For the given molecular weight (M_n and M_m), an increase in DOS will increase the adsorption of the lignosulfonate on the cement particle surface [101, 102]. However, with sufficiently large DOS value, the reverse will start to occur because of the increased water solubility that follows at the same time [39]. The DOS value does not account for other ionized groups that are also attached to the hydrophobic core of the lignosulfonate, namely the phenolic ($C_6H_5O^-$) and the carboxyl (COO^-) groups.

Table 4.2: The main characteristics of the lignosulfonates used in this thesis.

Polymer	DOS	RCL	M_n	M_m	PD	Ca	Na	Sugar
VHMW Na	≈37%	100%	41600	183000	4.40	-	-	0.05%
HMW Na	51%	85%	9900	84600	8.55	0.1%	6.9%	1.4%
HMW Ca	47%	85%	9900	74800	7.56	3.2%	1.1%	2.3%
LMW Na	64%	69%	7100	60900	8.58	0.4%	8.7%	-
LMW Ca	53%	66%	7400	73100	9.88	3.3%	0.2%	7.1%
LMWFS Ca	52%	54%	6800	54800	8.06	2.8%	0.1%	22.2%

The term RCL, in Table 4.2, designates **Relative Content of Lignin**. This value is relative to **VHMW Na**, which has the largest amount of lignin. As mentioned previously, this value increases under the process of ultrafiltration. The parameters M_n, M_m and PD $= M_\mathrm{m}/M_\mathrm{n}$ are the number average molecular weight, the mass average molecular weight and polydispersity, respectively [115]. More precisely, the terms M_n and M_m are defined with Equation 4.1 [115].

$$M_\mathrm{n} = \frac{\sum_\mathrm{i} N_\mathrm{i}\, M_\mathrm{w,i}}{\sum_\mathrm{i} N_\mathrm{i}} \qquad \wedge \qquad M_\mathrm{m} = \frac{\sum_\mathrm{i} N_\mathrm{i}\, M_\mathrm{w,i}^2}{\sum_\mathrm{i} N_\mathrm{i}\, M_\mathrm{w,i}} \tag{4.1}$$

The term N_i is the number of polymers with the molecular weight of $M_\mathrm{w,i}$. The physical unit used for the M_n, M_m and M_w is in [g/mol]. The method for determining the molecular weight values M_n and M_m, is called **Size Exclusion Chromatography** (**SEC**). Because of the sheer difficulty in measuring correct molecular weight, the values shown in the above table are not precise. However, the relative differences between those values are correct.

The last three columns in Table 4.2 are percentages, relative to the total weight of the (dry) lignosulfonate sample. The terms Ca and Na presents the amount of calcium

and sodium in the dry sample and sugar is the total sugar content, like hexose, pentose and others.

Adsorption Behavior of Lignosulfonates

Lignosulfonates can be classified as hydrophobic polyelectrolytes. This description refers to the fact that the polymer is dependent on its ionized groups for water solubility [39]. Lignosulfonates are in that respect similar to sulfonated naphthalene formaldehyde (**SNF**), which is also intrinsically a hydrophobic polymer [39]. The ionized groups (sometimes referred as the functional groups), attached to the hydrophobic core of the lignosulfonate, are basically the sulfonate (SO_3), the phenolic (C_6H_6O) and the carboxyl (COOH) groups [39]. In ionized state, they are as SO_3^-, $C_6H_5O^-$ and COO^-, respectively. These charged groups are distributed mainly on the polymer surface [105]. At pH = 7, the sulfonate group contributes to a negative charge of the lignosulfonate [69], while at pH $\geq$ 9, the phenolic and carboxylic groups contributes also to the (negative) surface charge density [38].

As shown in Table 4.2, each lignosulfonate type is very polydisperse (PD) with respect to molecular weight, i.e. each such polymer type consist of both small lignosulfonate molecules (order of magnitude 1 nm) and large lignosulfonate molecules (say, order of magnitude 100 nm). However, the amount of each fraction size is dependent on polymer type. For example, the **LMW Na** polymer consist more of the small sized lignosulfonate molecules, while the **VHMW Na** polymer consist more of the large sized lignosulfonates. This is shown with the schematic illustration to the left in Figure 4.3.

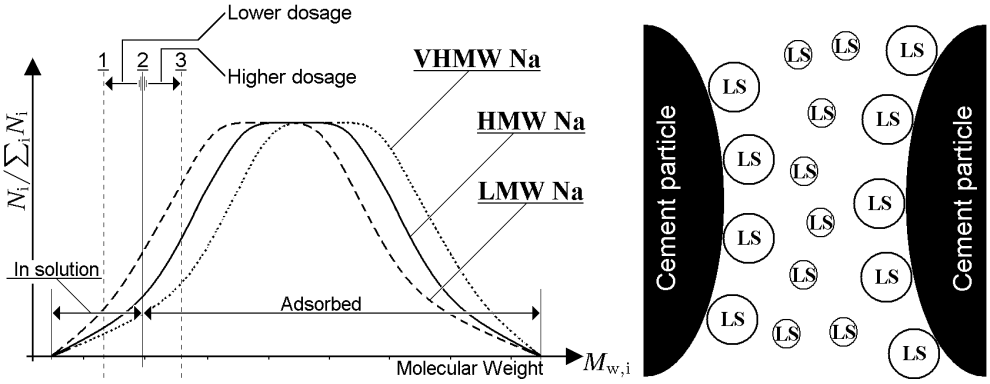


Figure 4.3: To the left: Schematic illustration of molecular weight distribution for the polymer types of **LMW Na**, **HMW Na** and **VHMW Na**. To the right: The larger polymer tend to adsorb to the cement surface, while the smaller ones tend to remain in the solution between the cement particles (dimensions are exaggerated).

The lignosulfonate molecules are heavily crosslinked and its typical dimension is reported to be in the order of 10 nm [18]. The larger molecular weight lignosulfonates can be looked upon as roughly spherical microgel [29, 84, 105], while the smallest fraction has more linear characteristics [84]. More precisely, as the molecular weight increases, the shape goes from a simple linear polymer (small M_w), to a branched polymer (medium M_w), to a more spherical microgel (large M_w) [84].

Investigations conducted by the scientists of Borregaard LignoTech, have gained a strong evidence of that the smaller lignosulfonate molecules remains in the

solution between the cement particles, while it is rather the larger lignosulfonate molecules that are adsorbed on the cement particle surface [39]. This is schematically shown with the right illustration of Figure 4.3 (dimensions are exaggerated). This means that for the given polymer dosage of **VHMW Na** applied to the suspension, a larger amount of lignosulfonate molecules are adsorbed to the surface of the cement particles, compared to when using an equal amount of **LMW Na**. This is shown with the vertical line marked with "**2**", in the left illustration. Also, when increasing the dosage of lignosulfonate, increased quantity of the large molecules start to appear in the solution. This is demonstrated with the dashed vertical line marked with "**3**". The opposite occurs when the dosage is reduced as shown with the dashed vertical line marked with "**1**".

Polymer dosage (or sometimes, dosage of lignosulfonate) is represented as [sbwc], and means "solid(s) by weight of cement". This value is given as percentage. For example, using 504 kg of FAC (see Table 4.6), with 0.6% sbwc of **HMW Na**, means that $504\,\text{kg}\cdot 0.006 \approx 3\,\text{kg}$ of dry lignosulfonate powder is used in the batch. The symbol $m_{\text{dp}}/m_{\text{c}}$ is sometimes used to designate the polymer dosage. The term m_{dp} is the mass of dry polymer and m_{c} is the mass of cement; i.e. $m_{\text{dp}}/m_{\text{c}} = 3\,\text{kg}/504\,\text{kg} \approx 0.6\%$.

4.2.3 Aggregates

The aggregates used in this project are supplied by NorStone AS, which is part of the German cement and building materials group Heidelberger Zement AG. The aggregates are taken from one and the same geological formation in Årdal, Norway. The formation originates from a moraine and is sorted by a relatively short distance by melt water from the corresponding glacier. At least 70% of the $0-8$ mm aggregates consists of uncrushed materials. Aggregates larger than 8 mm consists of about 30% of natural and about 70% of crushed materials. The density of the aggregate sample is $\rho_{\text{a}} = 2670\,\text{kg/m}^3$.

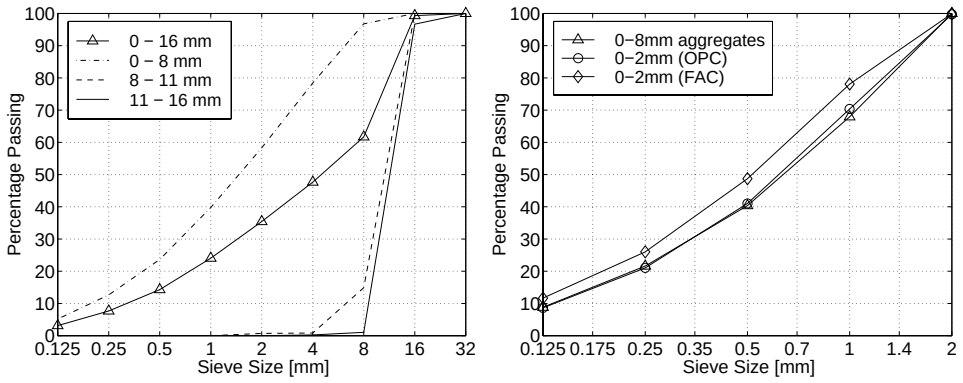


Figure 4.4: The aggregate grading as percentage passing versus sieve size. To the left is the grading curve used for the concrete mixes. To the right are the grading curve of the mortars.

The grading curves are shown in Figure 4.4. To the left is the grading used for all the concrete batches, and to the right are the grading curves for the mortars. The "$0-8$ mm aggregates"-grading curve shown in the right illustration, is the $0-2$ mm part of the $0-8$ mm aggregates used in the concrete mix. As will be apparent shortly, it must be very similar to the $0-2$ mm aggregates used in the OPC-mortar, marked

as "$0-2$ mm (OPC)". The "$0-2$ mm (FAC)"-grading curve shown, is the aggregate grading used for the fly ash mortars.

The moisture content for the $0-8$ mm aggregates are kept at $3.5\% \pm 0.5\%$. The $8-16$ mm aggregates are bone dry, i.e. measured with roughly 0% moisture. The moisture content of the $0-2$ mm aggregates are held at $2.7\% \pm 0.1\%$. The absorbing capability of the $0-2$ mm, $0-8$ mm and $8-16$ mm aggregates are 0.8%, 0.8% and 0.5% respectively (values are supplied by NorStone). It is assumed that absorbed water is not part of the cement paste inside the concrete, or inside the mortar.

4.3 Mix Design

Because of how rheological measurements on concrete requires a large resources in terms of test material, workforce and time, it is not practical to apply the whole test program to concrete. Rather, the mix design of $0-2$ mm mortar, inside the concrete, is calculated and thereafter used to a much larger test program. This is done in the hope that the corresponding mortar batches will simulate the rheological behavior of the concrete. By this approach, it possible to create a larger and more complicated test program, to investigate the effects of the different lignosulfonate types.

A similar approach to the above, has been made by Mørtsell [80, 81]. There, instead of calculating a $0-2$ mm mortar from the concrete, a $0-0.125$ mm filler modified cement paste is rather calculated. There are two reasons for the use of mortar rather than filler modified cement paste in this thesis. First, the dispersing mechanism from the $0-2$ mm aggregates on the clumped and coagulated cement particles is more similar to what applies from the $0-16$ mm aggregates in concrete. Secondly, making a filler modified cement paste at $w/c = 0.5$ and 0.6 with the concomitant use of plasticizers, generates a *soup* with too low shear viscosity η for the ConTec Viscometer 4 to register.

4.3.1 Mix Design of Concrete (OPC)

When making the mix design for the concrete, it was preferred to have about the same slump value (see Section 4.4.2) for the most of the mixes containing the different polymer types. This was preferred when comparing their difference in workability and workability retention. The goal was to gain an initial slump value around and above 200 mm (see for example Figure 6.5, Page 108). For a given polymer dosage, the amount of binder[2] was increased until the requested slump value was gained.

Initially, a higher dosage of lignosulfonate and lower amount of binder was used in the mix design. This was done to simulate a more realistic and economical concrete mix. The concrete came out well in respect to flowability and stability. However, the corresponding $0-2$ mm mortar batch had bleeding problems and also segregated. The solution to the instability problem of the mortar was to make it more viscous, by reducing the amount of polymer dosage. However, to maintain the slump value of 200 mm at the same time for the concrete, a larger amount of binder had to be used. The final mix design for the concrete at $w/c = 0.4, 0.5$ and 0.6 is shown in Table 4.3.

The amount of aggregates shown in Table 4.3 are in (bone) dry condition. The $\dot{\gamma}_m/\dot{\gamma}$-ratio is explained in Section 4.4.1. The mortar content shown in the table, assumes an air content of 2%. This value is generally measured (see Section 5.5.1).

[2] **Binder** means the combined mix of cement, water and dry polymer (i.e. cement paste).

Table 4.3: Mix design for the concrete at the different w/c-ratios on the $1\,\mathrm{m}^3$ basis.

w/c	Binder	Mortar	Cement	$m_{\mathrm{dp}}/m_{\mathrm{c}}$	Aggregates	Density	$\frac{\dot{\gamma}_{\mathrm{m}}}{\dot{\gamma}}$
0.4	320 l	574 l	442 kg	0.6%	1762 kg	2396 kg/m^3	4.0
0.5	305 l	564 l	371 kg	0.3%	1802 kg	2372 kg/m^3	4.1
0.6	295 l	558 l	320 kg	0.1%	1829 kg	2354 kg/m^3	4.2

4.3.2 Calculating Mix Design of Mortar (OPC)

In the following is a short demonstration of how the $0-2\,\mathrm{mm}$ mortar is calculated from the concrete mix, at the w/c-ratio of 0.4. The same calculation scheme applies for the other two mix design at $w/c = 0.5$ and 0.6. This calculation is made relative to one cubic meter of batch. The resulting mix design is shown in Table 4.4.

As shown in the left illustration of Figure 4.4, the percentage passing at the 2 mm sieve size is 35.4%. Therefore, the amount of $0-2\,\mathrm{mm}$ aggregates in the concrete is about $0.354 \cdot 1762\,\mathrm{kg} = 624\,\mathrm{kg}$ (see Table 4.3). This corresponds to a volume of $624\,\mathrm{kg}/\rho_{\mathrm{a}} = 234\,\mathrm{l}$. Now, from Table 4.3, the amount of binder in the concrete is 320 l. With this, the volume of mortar inside the concrete is calculated as $234\,\mathrm{l} + 320\,\mathrm{l} + 20\,\mathrm{l} = 574\,\mathrm{l}$ (the volume of 20 l corresponds to 2% air, per cubic meter of concrete batch). Hence, the relative amount of binder in the $0-2\,\mathrm{mm}$ mortar (inside the concrete) is $320\,\mathrm{l}/574\,\mathrm{l} = 55.8\%$. This means that in the mix design for the mortar, the amount of binder must be 558 l on the $1\,\mathrm{m}^3$ basis (shown in Table 4.4). With $0.558\,\mathrm{m}^3 = m_{\mathrm{c}}/\rho_{\mathrm{opc}} + m_{\mathrm{w}}/\rho_{\mathrm{w}} + V_{\mathrm{dp}} = m_{\mathrm{c}}/\rho_{\mathrm{opc}} + 0.4 \cdot m_{\mathrm{c}}/\rho_{\mathrm{w}} + 0.006 \cdot m_{\mathrm{c}}/\rho_{\mathrm{dp}}$ and then solving for m_{c} gives $m_{\mathrm{c}} = 770\,\mathrm{kg/m}^3$. The terms V_{dp} and ρ_{dp} are the volume and density of dry lignosulfonate polymer, respectively. Since the air content of concrete is about 2%, the air content of mortar should be $20\,\mathrm{l}/574\,\mathrm{l} = 3.48\%$ or 34.8 l. This means that the amount of $0-2\,\mathrm{mm}$ aggregates is $(1000\,\mathrm{l} - 558\,\mathrm{l} - 34.8\,\mathrm{l}) \cdot \rho_{\mathrm{a}} = 1087\,\mathrm{kg}$. Hence, density is $770 + 0.4 \cdot 770 + 0.006 \cdot 770 + 1087 + 0.008 \cdot 1087 = 2178\,\mathrm{kg/m}^3$. The value $0.6\% \cdot 770\,\mathrm{kg}$ is the mass of dry polymer and $0.8\% \cdot 1087\,\mathrm{kg}$ is the mass of absorbed water, which is added to the already supplied water of $0.4 \cdot 770\,\mathrm{kg}$.

Table 4.4: Mix design for mortar at the different w/c-ratios on the $1\,\mathrm{m}^3$ basis (OPC).

w/c	Binder	Cement	$m_{\mathrm{dp}}/m_{\mathrm{c}}$	Aggregates	Density	$m_{\mathrm{a}}/m_{\mathrm{c}}$
0.4	558 l	770 kg	0.6%	1087 kg	2178 kg/m^3	1.4
0.5	541 l	658 kg	0.3%	1132 kg	2130 kg/m^3	1.7
0.6	529 l	574 kg	0.1%	1162 kg	2090 kg/m^3	2.0

The $m_{\mathrm{a}}/m_{\mathrm{c}}$-ratio shown in Table 4.4, describes the mass ratio of aggregates to cement. This ratio can give a measure of grinding and dispersing effect from the aggregates on the clumped and coagulated cement particles. A higher value means a higher effect. This value is used in Chapter 5, when discussing reproducibility.

4.3.3 Mix Design of Mortar (FAC)

To begin with, the mix design of Table 4.4 was used when applying the FAC. The resulting mix design for the FAC-case is shown in Table 4.5, and is constructed with the same amount of binder as a constraint, namely 558 l.

As is discussed in Section 5.4, due to reproducibility problems when using the

Table 4.5: Old mix design for mortar using the FAC.

w/c	Binder	Cement	$m_{\mathrm{dp}}/m_{\mathrm{c}}$	Aggregates	Density	$m_{\mathrm{a}}/m_{\mathrm{c}}$
0.4	558 l	751 kg	0.6%	1088 kg	2153 kg/m^3	1.4

FAC, a different mix design is applied. This mix design is shown in Table 4.6. With this change, a different batch of sand (from a specific big-bag) is used. The grading of this sand is shown in Figure 4.4 and is marked as "$0-2$ mm (FAC)". For further information about the aggregates, see Section 4.2.3.

Table 4.6: New mix design for mortar using the FAC.

w/c	Binder	Cement	$m_{\mathrm{dp}}/m_{\mathrm{c}}$	Aggregates	Density	$m_{\mathrm{a}}/m_{\mathrm{c}}$
0.57	460 l	504 kg	0.6%	1388 kg	2193 kg/m^3	2.7
0.57	460 l	505 kg	0.3%	1388 kg	2193 kg/m^3	2.7

4.4 Test Methods

4.4.1 ConTec Viscometers

Verification of Viscometers

Before the experimental part started it was necessary to verify that the ConTec **BML** Viscometer 3 and the ConTec Viscometer 4 viscometer were working properly. The verification is made on three frontiers. First check if the load cell (that registers the torque $\hat{T}$, c.f. Equation 3.24 on Page 59) is working properly by applying fixed loads directly to it and see if the software FreshWin[3] registers the correct value. The second check consist of verifying that the rotational frequency of the outer cylinder f_{o} corresponds to what the software is registering. The third check consists of measuring the shear viscosity η of a high viscous Newtonian fluid with known viscosity value and verifying that the viscometer measures the correct value. The above verification procedure is applied both to the ConTec **BML** Viscometer 3 and ConTec Viscometer 4.

Shear Rate of Concrete Inside the ConTec BML Viscometer 3

The aim is to make the magnitude of shear rate $\dot{\gamma}$ for a concrete batch inside the ConTec **BML** Viscometer 3, correspond to what applies in reality. During casting (see Figure 3.2 on Page 54), it is not unexpected if about 5 cm thick layer of concrete is flowing with the speed ranging from about 0.1 to 0.5 m/s. With this, the shear rate is in the order of magnitude from $\dot{\gamma} \propto (10\,\mathrm{cm/s})/(5\,\mathrm{cm}) \approx 2\,\mathrm{s}^{-1}$ to $\dot{\gamma} \propto (50\,\mathrm{cm/s})/(5\,\mathrm{cm}) \approx 10\,\mathrm{s}^{-1}$. Tests in the viscometer are made from $f_{\mathrm{min}} = 0.1$ rps to $f_{\mathrm{max}} = 0.5$ rps and since the thickness of the concrete layer in this device is $\Delta R = R_{\mathrm{o}} - R_{\mathrm{i}} = 14.5\,\mathrm{cm} - 10\,\mathrm{cm} = 4.5\,\mathrm{cm}$, the order of magnitude for shear rate is from $\dot{\gamma} \propto (2\pi \cdot f_{\mathrm{min}} \cdot R_{\mathrm{o}})/\Delta R \approx 2\,\mathrm{s}^{-1}$ to $\dot{\gamma} \propto (2\pi \cdot f_{\mathrm{max}} \cdot R_{\mathrm{o}})/\Delta R \approx 10\,\mathrm{s}^{-1}$. Hence, the values of f_{min} and f_{max} are considered to be correct. These values are shown in Table 3.1 (Page 64).

[3]The software that controls the ConTec viscometers is named FreshWin.

Shear Rate of Mortar Inside the ConTec Viscometer 4

Since the OPC-mortar is supposed to represent the $0-2$ mm mortar inside the concrete (see Section 4.3.2), it is only natural to have the same shear rata condition inside the ConTec Viscometer 4, as applies for the mortar inside a concrete. To calculate the shear rate $\dot{\gamma}_{\mathrm{m}}$ for the mortar inside the concrete, one must make the use of Figure 2.6 (Page 22). In this figure, the black and white particles represent the larger cement particles. Extrapolate this figure to concrete, the black and white particles now represent the $2-16$ mm aggregates. The material in-between is now the $0-2$ mm mortar. This configuration is shown in Figure 4.5. Illustration **A** is the same as illustration **C** in Figure 2.6. Illustration **B** (in Figure 4.5) demonstrates how the mortar layer goes under a larger shear rate (which is $\dot{\gamma}_{\mathrm{m}}$) than the concrete as a whole (which is $\dot{\gamma}$). The term δD designates the average thickness of the mortar layer and D the average diameter of the $2-16$ mm aggregates.

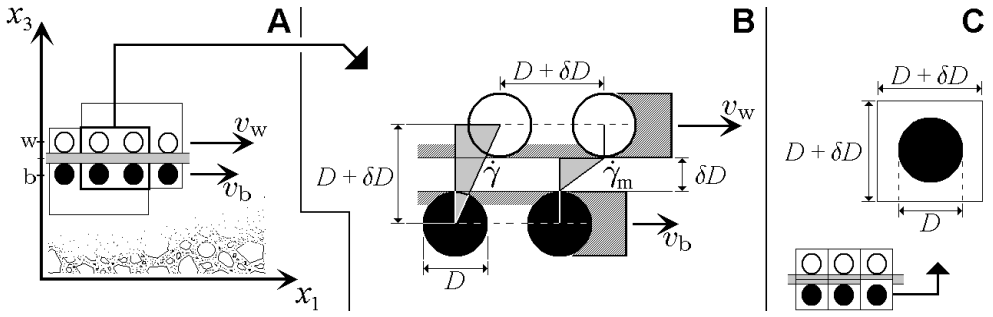


Figure 4.5: Shear rate condition of mortar, inside the concrete. Illustration **A** is the same as illustration **C** in Figure 2.6 (Page 22). Illustration **B** demonstrates in a more detail how the shear rate of mortar $\dot{\gamma}_{\mathrm{m}}$ is larger than of the concrete as a whole $\dot{\gamma}$. Illustration **C** represents a "repeating unit" of the concrete sample.

With the assumed velocity profile of $\mathbf{v} = v_1(x_3,t)\mathbf{i}_1$ for the concrete, the shear rate develop into $\dot{\gamma} = |\partial v_1/\partial x_3| = \partial v_1/\partial x_3 \geq 0$. As such, $\dot{\gamma}$ can be approximated with $(v_{\mathrm{w}} - v_{\mathrm{b}})/(D+\delta D)$. Likewise, the shear rate of the mortar (inside the concrete) can be approximated with $(v_{\mathrm{w}} - v_{\mathrm{b}})/\delta D$. This gives the following equation:

$$\frac{D+\delta D}{\delta D} = \frac{\frac{v_{\mathrm{w}}-v_{\mathrm{b}}}{\delta D}}{\frac{v_{\mathrm{w}}-v_{\mathrm{b}}}{D+\delta D}} \approx \frac{\dot{\gamma}_{\mathrm{m}}}{\dot{\gamma}} \tag{4.2}$$

Illustration **C** in Figure 4.5 demonstrates a "repeating unit" of a concrete sample. This leads to the relationship of $D^3/(D+\delta D)^3 \propto V_{\mathrm{a}}/V_{\mathrm{t}}$, where V_{a} is the volume of the $2-16$ mm aggregates (inside the concrete) and V_{t} is the total volume of the concrete sample. With this, Equation 4.2 can be rewritten to the following:

$$\frac{\dot{\gamma}_{\mathrm{m}}}{\dot{\gamma}} \approx \frac{D+\delta D}{\delta D} = \frac{1}{1-\frac{D}{D+\delta D}} \propto \frac{1}{1-(V_{\mathrm{a}}/V_{\mathrm{t}})^{\frac{1}{3}}} \tag{4.3}$$

The $\dot{\gamma}_{\mathrm{m}}/\dot{\gamma}$-ratio is shown in the last column in Table 4.3. There it is demonstrated that the condition of $\dot{\gamma}_{\mathrm{m}}/\dot{\gamma} \propto 4$ applies, meaning that the same order of magnitude of shear rate should apply for both concrete and mortar, but with the constraint of that the $\dot{\gamma}_{\mathrm{m}}$ value should be larger than of $\dot{\gamma}$.

Tests in the ConTec Viscometer 4 are made from $f_{\min} = 0.05$ rps to $f_{\max} = 0.45$ rps. Since the thickness of the mortar layer inside the viscometer is $\Delta R =$

$R_o - R_i = 10.1\,\text{cm} - 8.5\,\text{cm} = 1.6\,\text{cm}$, the order of magnitude for shear rate is from $\dot{\gamma}_m \propto (2\pi \cdot f_{min} \cdot R_o)/\Delta R \approx 5\,\text{s}^{-1}$ to $\dot{\gamma}_m \propto (2\pi \cdot f_{max} \cdot R_o)/\Delta R \approx 20\,\text{s}^{-1}$. This means that the condition mentioned in the previous paragraph applies. Therefore, the values of f_{min} and f_{max} can be considered to be relatively correct. These values are shown in Table 3.1 (Page 64).

Tests With the ConTec Viscometers

How torque $\hat{T}$ is generated and converted into plastic viscosity μ and yield value τ_o is discussed in Section 3.4. In fact, the whole Chapter 3 is devoted to the description of ConTec viscometers, relative to the Bingham fluid. For example, with Figures 3.6 and 3.7 (Page 60), a step by step description is made of how the continuous torque data is filtered to create a set of discrete torque points $\hat{T}$, which is then used to calculate the slope H and the point of intersection with the ordinate G, giving $\hat{T} = H\,f_o + G$. Using Equations 3.26 and 3.27, the G and H values are converted to the yield value τ_o and the plastic viscosity μ.

4.4.2 Slump

Rheological behavior can be determined, to some degree, by the (Abrams) slump test. This method is used extensively in site work all over the world. The slump test is prescribed by ASTM C 143-90a and BS 1881:Part 102:1983. Additional information about the slump test can for example be found in [10, 24, 88].

Because of its simplicity, the slump test remains the most widely used test for characterizing concrete consistency. As shown in Figure 4.6, the apparatus consist of a mould in the shape of a truncated metal cone, open at both ends. The internal diameter of the slump cone, is 200 mm at the base, 100 mm diameter at the top and has a height of 300 mm. This device is usually provided with foot pieces and handles.

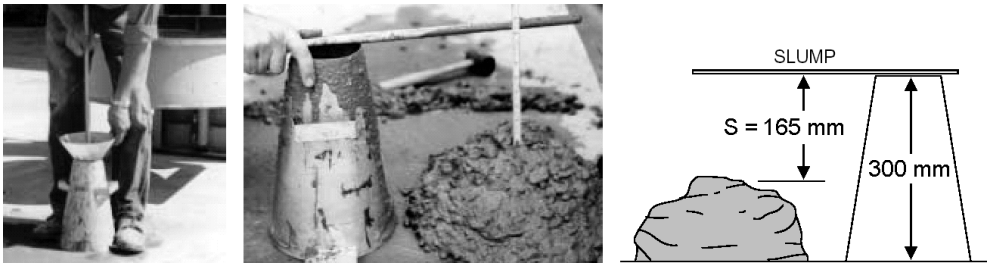


Figure 4.6: Slump measurement is made by placing the slump cone beside the slumped sample and laying the tamping rod across the cone, extending over the specimen. Then the distance from the bottom of the rod to the slumped concrete, gives the slump value S (from [24]).

Figure 4.6 demonstrates how the slump is measured. Basically the procedure consist of filling the metal cone with concrete, by a specific procedure (see the left illustration). Thereafter, the metal cone is lifted, leaving the concrete sample behind which "slumps down" by the action of gravity $\mathbf{g}$. This is shown with the center and right illustrations. Being a viscoplastic fluid, the slump material stops flowing downward when the condition $(\mathbf{T} : \mathbf{T})/2 \leq \tau_o^2$ becomes dominated (see Equation 3.5, Page 53). Hence, it is not unexpected that there is a strong relationship between the slump value S and the true yield value τ_o of the concrete sample. A relationship

between the slump value S and the yield value τ_o measured with the ConTec **BML** Viscometer 3, is presented in Section 6.3.1.

4.5 Test Program

The purpose of the test program is presented in Section 1.2.1. The test program consists of testing the effects of seven different types of plasticizers/superplasticizers. Six of them are lignosulfonates produced by Borregaard LignoTech, described in Section 4.2.2. The other type of superplasticizer is a Naphthalene based product, namely Sulfonated Naphthalene Formaldehyde (**SNF**). The **SNF** has the commercial name Suparex M40 and is produced by Hodgson Chemicals Ltd. In addition to this, a batch without any plasticizers/superplasticizers is always included when conducting a mortar measurement. Such a batch is designated as **Without P/SP**. Both the **SNF** and **Without P/SP** serve mainly as a reference, in comparison to the effects of lignosulfonates. From the results of this test program, other investigations are also made, like the comparison of slump results with yield values τ_o, measured by the ConTec **BML** Viscometer 3.

The above polymer types are used in mortar of two different cement types, namely the OPC and FAC. A description about these cement types is made in Section 4.2.1. However, no concrete batch is mixed or measured with FAC. This is shown in Table 4.7, where only the "OPC-boxes" are marked with a "X". A box with a X mark, means that the specific batch is mixed and measured. An empty box means that no measurement is made of the corresponding mix. This applies to all tables shown in Section 4.5.

The test program is basically divided between the types of cement used, or more precisely between the OPC and FAC. For the OPC-case, three different w/c-ratios are used, namely of 0.4, 0.5 and 0.6. For each w/c-ratio, a specific dosage of plasticizing polymer is applied. A different scheme is valid for the FAC-case. There the same w/c-ratio is always applied ($w/c = 0.57$), however with a different polymer dosage, either of 0.6% sbwc or of 0.3% sbwc. The mix design for the cases of concrete (OPC) and mortar (OPC, FAC) is presented in Tables 4.3, 4.4 and 4.8 (Page 78).

Table 4.7: Concrete measurements at 23°C (Viscometer and Slump).

Cement type →	OPC	OPC	OPC	FAC	FAC
Polymer [sbwc] →	0.6%	0.3%	0.1%	0.6%	0.3%
w/c →	0.4	0.5	0.6	0.57	0.57
VHMW Na	X	X	X		
HMW Na	X	X	X		
HMW Ca	X	X	X		
LMW Na	X	X	X		
LMW Ca	X	X	X		
LMWFS Ca	X	X	X		
SNF	X	X	X		
Without P/SP					

Table 4.8: Mortar measurements at 5°C.

Cement type →	OPC	OPC	OPC	FAC	FAC
Polymer [sbwc] →	0.6%	0.3%	0.1%	0.6%	0.3%
w/c →	0.4	0.5	0.6	0.57	0.57
VHMW Na					
HMW Na	X	X	X		
HMW Ca	X				
LMW Na	X	X	X		
LMW Ca					
LMWFS Ca					
SNF	X	X	X		
Without P/SP	X	X	X		

Table 4.9: Mortar measurements at 23°C.

Cement type →	OPC	OPC	OPC	FAC	FAC
Polymer [sbwc] →	0.6%	0.3%	0.1%	0.6%	0.3%
w/c →	0.4	0.5	0.6	0.57	0.57
VHMW Na	X	X	X	X	X
HMW Na	X	X	X	X	X
HMW Ca	X	X	X		
LMW Na	X	X	X	X	X
LMW Ca	X	X	X	X	X
LMWFS Ca	X	X	X	X	X
SNF	X	X	X	X	X
Without P/SP	X	X	X	X	X

Table 4.10: Mortar measurements at 38°C.

Cement type →	OPC	OPC	OPC	FAC	FAC
Polymer [sbwc] →	0.6%	0.3%	0.1%	0.6%	0.3%
w/c →	0.4	0.5	0.6	0.57	0.57
VHMW Na	X	X		X	X
HMW Na	X	X		X	X
HMW Ca	X	X			
LMW Na	X	X		X	
LMW Ca	X	X		X	
LMWFS Ca	X	X		X	
SNF	X	X		X	X
Without P/SP	X	X			

Chapter 5

Mixing Procedure and Reproducibility

5.1 Introduction

In this thesis, a number of four, and sometimes three, repeated batches of the same mix design are measured with the ConTec **BML** Viscometer 3, ConTec Viscometer 4 and the Abrams cone. Thereafter, comparisons of the rheological values are made, to see if the same outcome is reproduced. This is always done before actually starting with the measurements of interest and is intended to provide a (rough) error estimation to the whole experiment. However, this is not the only reason. The second objective is to reveal and compensate for different systematic errors that are always present in this type of experimental work. The specific rheological behavior shown in this chapter is not discussed until in Chapter 6.

In the progress of time, the reproducibility effort became a larger and larger part of this thesis and hence is the subject of its own chapter.

5.2 Concrete Measurements (OPC)

5.2.1 Mixing and Measuring Procedure

A 50 liters mixer from Maschinenfabrik Gustav Eirich of type SKG1 is used in (re)mixing of the concrete batches, prior to a rheological measurement. This mixer is shown in the left illustration of Figure 5.1. Before mixing, the $0-16$ mm aggregates (beginning with the $11-16$ mm ones) and then the cement, are pored into the mixing bowl. The lignosulfonate, the air-detraining agent[1] TBEP and water, are premixed before adding. It is a common experience that the plasticizing polymers are more effective if added a few minutes after water addition (see for example [131]). To have the mixing procedure for the different batches as equal as possible, such maneuver is not made.

For all mixes with the Norcem Standard Cement (OPC), the mass ratio of air-detraining agent versus polymer (i.e. the $m_{\text{tbep}}/m_{\text{dp}}$-ratio) is kept at 1%. This is a magnitude that is generally applied [105]. This value is also used when making

[1] Tributoxyethyl phosphate $\Rightarrow (C_4H_9OC_2H_4O)_3PO$

a batch of mortar or cement paste using the OPC. However, the $m_{\text{tbep}}/m_{\text{dp}}$-ratio is increased up to 1.5% for batches containing the NORCEM FLY ASH CEMENT (FAC).

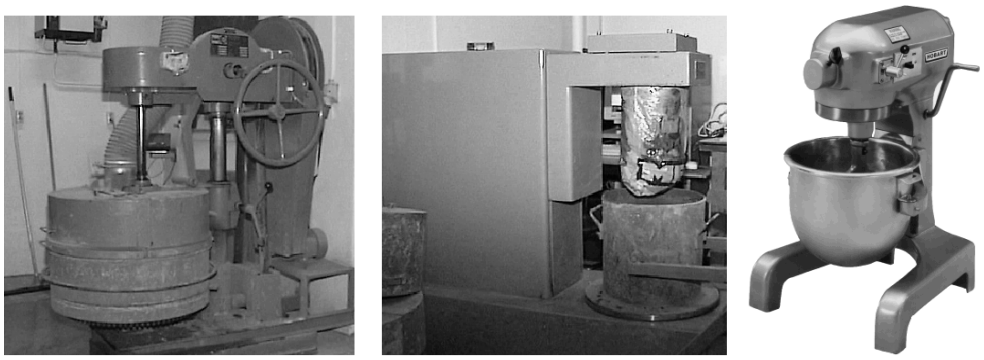


Figure 5.1: To the left is the GUSTAV EIRICH concrete mixer and in the middle is the CONTEC **BML** VISCOMETER 3 with a plastic bag covering its inner cylinder. To the right is the HOBART AE120 mixer and is used when mixing mortar or cement paste.

The **mixing procedure** and the **measuring procedure** for the concrete batches are given in Box 1 and 2, respectively. The mix design is shown in Table 4.3 (Page 78). The variable t designates the time after water addition.

Box 1: Mixing procedure for concrete batches.

1. Dry mixing of the $0-16\,\text{mm}$ aggregates and the cement for one minute, in the time period of $t \in [-1\,\text{min}, 0\,\text{min}]$.
2. Wet mixing, in the time period of $t \in [0\,\text{min}, 1\,\text{min}]$. The time "$t = 0\,\text{min}$" is defined as the initial start of water addition. The water (which is premixed with lignosulfonate) is pored into the mix in the time interval of about 30 seconds.
3. Rest, in the time period of $t \in [1\,\text{min}, 3\,\text{min}]$ (i.e. for two minutes).
4. Mixing, in the time period of $t \in [3\,\text{min}, 5\,\text{min}]$ (i.e. for two minutes).

Box 2: Measuring procedure for concrete batches.

1. $t = 10\,\text{min}$: Rheological measurement with the concrete viscometer.
2. $t = 34\,\text{min}$: Remixing for 2 minutes in the concrete mixer.
3. $t = 40\,\text{min}$: Rheological measurement with the concrete viscometer.
4. $t = 64\,\text{min}$: Remixing for 2 minutes in the concrete mixer.
5. $t = 70\,\text{min}$: Rheological measurement with the concrete viscometer.
6. $t = 94\,\text{min}$: Remixing for 2 minutes in the concrete mixer.
7. $t = 100\,\text{min}$: Rheological measurement with the concrete viscometer.

After each measurement with the CONTEC **BML** VISCOMETER 3, the concrete is placed back into the mixer to rest there. The resting time consist of less than 23 minutes until remixing occurs. Care is taken to ensure a minimum loss of moisture (due to evaporation) for this time period (see Figure 5.1, to the left). Six minutes before making the second measurement on the same batch (at $t = 40\,\text{min}$), the concrete is remixed for two minutes. Thereafter, the concrete is under manual agitation during

the process of moving it from the mixing device to the viscometer. This applies also before the measurements conducted at $t = 70$ min and $t = 100$ min.

After each measurement, a rough cleaning procedure is applied to the inner and outer cylinder of the viscometer. This consisted of removing the remaining concrete with manual force. Thereafter, the inner cylinder is covered with a plastic bag, to inhibit a moisture loss from the remaining fine mortar, stuck on this cylinder. This is shown with the center illustration of Figure 5.1. A plastic cover is also placed on top of the outer cylinder, i.e. on the top of the bucket of the viscometer (see the center illustration of Figure 3.5, Page 56).

5.2.2 Quest for Reproducibility

With the verification procedure described in Section 4.4.1, it is confirmed that the viscometers are working correctly. This does however not ensure a correct viscometric measurement. Errors related to the mixing and measuring procedure can be present. Measuring the same rheological results from the different batches of the same mix design, indicate that such errors are at minimum. To test if such reproducibility exists, then for each mix design at $w/c = 0.4, 0.5$ and 0.6, number of four repeated batches are made and measured. The results of these tests are shown in Figures 5.2, 5.3 and 5.4.

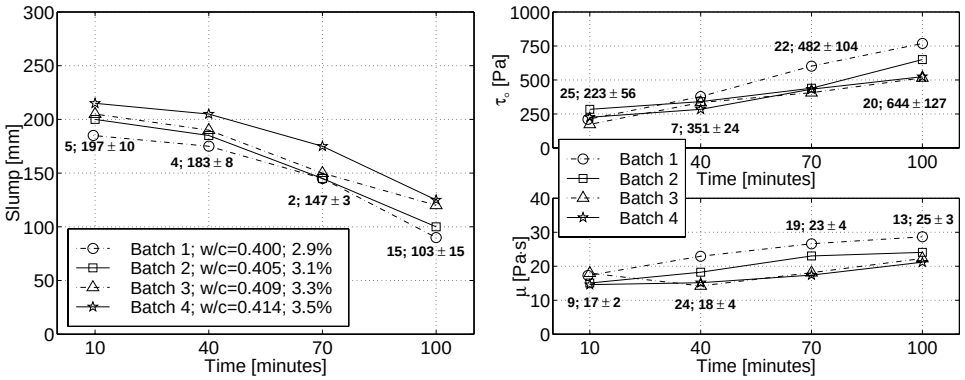


Figure 5.2: Reproducibility results for concrete at $w/c = 0.4$, using **HMW Na**. The sequence of numbers represents: Coefficient of variation; Average ± Standard deviation. To the left is the slump value and to the right is the yield value τ_o (top) and plastic viscosity μ (bottom).

Due to some experimental error, the water content in the batch number 2, 3 and 4 of $w/c = 0.4$ [**HMW Na**; 0.6% sbwc], are larger than corresponds to the target value of $w/c = 0.4$. Estimated w/c-ratio for these batches are shown in Figure 5.2. The type of error that is responsible for this, is only present for these three batches.

The sequence of numbers shown in Figures 5.2 to 5.4 are the **coefficient of variation**[2] of the measurements and the **average ± standard deviation**. These values are frequently applied in Chapter 6, when representing the various rheological results. There, the average value is always used as the actual value for the **HMW Na**-batch, and its standard deviation is used in calculating the 95% confidence interval for that particular case, based on t-distribution (see Section 6.2.2). As shown for example in Figures 6.2 and 6.4 (Page 105), the confidence interval is plotted with four vertical

[2]Coefficient of variation = 100 · standard deviation/average (is expressed as a percentage) [133].

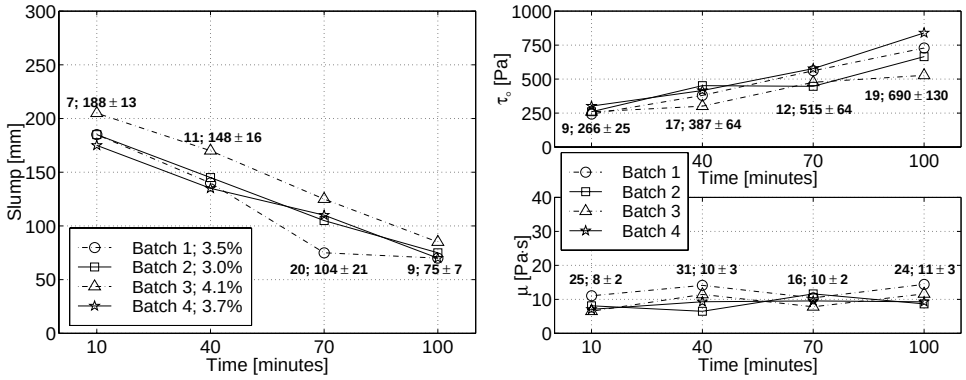


Figure 5.3: Reproducibility results for concrete at $w/c = 0.5$, using **HMW Na**. To the left is the slump value and to the right is the yield value τ_o (top) and plastic viscosity μ (bottom).

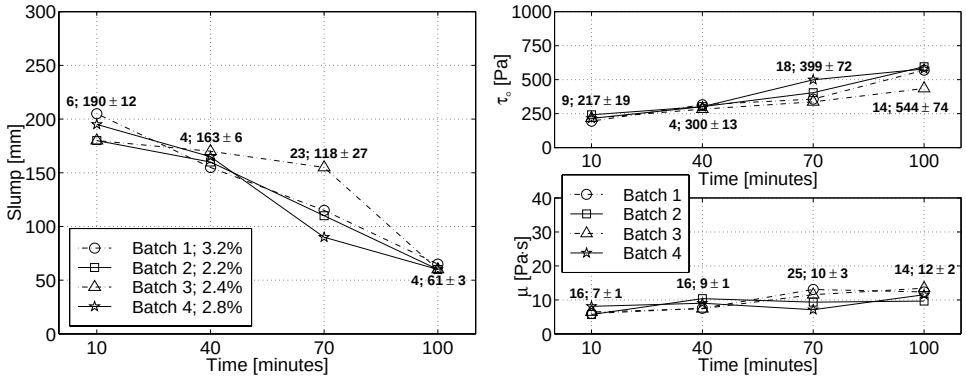


Figure 5.4: Reproducibility results for concrete at $w/c = 0.6$, using **HMW Na**. To the left is the slump value and to the right is the yield value τ_o (top) and plastic viscosity μ (bottom).

lines. In Figure 5.2, the batch number 4 is excluded when calculating the coefficient of variation, average and standard deviation, as a consequence of the uncontrolled increase in w/c-ratio, discussed just previously.

The rheological results of the concrete batches shown in Figures 5.2 to 5.4, are more or less overlapping each other. For the measured yield value τ_o and plastic viscosity μ, the coefficient of variation is roughly between 10 and 20%. For the slump results, this coefficient is lower, or roughly between 5 and 10%. As such, the mixing and measuring procedure of Box 1 and 2 can be considered to be acceptable. However, regardless of this result, what is most important in Figures 5.2 to 5.4, are the values of standard deviation. As mentioned above, they are used in the calculation of the 95% confidence interval, which again demonstrates the degree of significance between the different rheological results shown in Chapter 6.

It is always the **HMW Na** polymer that is applied in the reproducibility tests shown here. The reason for this is related to the importance of this polymer. It was early pointed out by Borregaard LignoTech, that this lignosulfonate is one of the most important one, of the types given to this work. As such, the **HMW Na** polymer is also used in the following sections, when determining the reproducibility for the

mortar batches. Because of how it requires large resources in terms of test material, workforce and time, it is only for the **HMW Na**-case that the reproducibility is measured. It would have enhanced this work if reproducibility were also tested using other types of lignosulfonates. Since such data is not available, the 95% confidence interval for the **HMW Na**-case is sometimes used as a rough error estimation for other lignosulfonate cases. Fortunately, it is possible to calculate a 95% confidence interval based on a single measurement. This is done for all batches of this work (see for example Figure 6.2, Page 105). The calculation is explained in Section 6.2.2 and is in accordance with a textbook by Tattersall and Banfill [128]. This maneuver gives a clearer picture of the degree of significance between the different rheological results, shown in Chapter 6. This consideration applies also for the mortar case.

5.3 Mortar Measurements (OPC)

5.3.1 Mixing Procedure

A Hobart AE120 mixer is used for all the mortar mixes. This device is shown in the right illustration of Figure 5.1. It has three speed settings for the agitator and the attachment (i.e. the whips) respectively: **Speed 1**: 106 and 61 rpm, **Speed 2**: 196 and 113 rpm and **Speed 3**: 358 and 205 rpm. The agitator rotates clockwise and the attachment rotates counter clockwise. The attachment is a blade shaped (like an open shield). Before mixing takes place, the $0-2\,\text{mm}$ aggregates and then the cement, is pored into the mixing bowl. As when mixing a concrete batch, the lignosulfonate, air-detraining agent TBEP and water are premixed before addition to the mix of $0-2\,\text{mm}$ aggregates and cement. The **mixing procedure** for the mortar is currently nominally the same as the one used for the concrete:

Box 3: Mixing procedure for mortar batches (old).

1. Dry mixing of the $0-2\,\text{mm}$ aggregates and the cement at speed 1, in the time period of $t \in [-1\,\text{min}, 0\,\text{min}]$.
2. Wet mixing at speed 1, in the time period of $t \in [0\,\text{min}, 1\,\text{min}]$. At the time $t = 0\,\text{min}$ is the start of water addition. The water (which is premixed with lignosulfonate) is pored into the mix in the time interval of ca. 45 to 50 seconds. Pouring all the water into the mix at a shorter time, gave a less efficient mixing. In such cases, the formation of one (or more) big cement clump became a reality, with about one third of the cement still not wetted.
3. Rest and hand mixing, for 2 minutes ($t \in [1\,\text{min}, 3\,\text{min}]$).
4. Mixing at speed 2, for 2 minutes ($t \in [3\,\text{min}, 5\,\text{min}]$).

The hand mixing in Item 3 above, is introduced in order to brake up slam layers and coarsely coagulated cement clumps that resided at the bottom in the bowl. The total mixing time, including dry mixing is six minutes. The anticipation was that the above mixing procedure would be sufficient enough, since the $0-2\,\text{mm}$ aggregates were expected to brake up any remaining small cement clumps, and then disperse the residual of coagulated cement particles in an adequate manner.

5.3.2 Measuring Procedure

After each measurement with the ConTec Viscometer 4, the mortar is placed back into the bowl of the Hobart mixer, to rest there. Care is taken to ensure a minimal

loss of moisture (due to evaporation). Six minutes before repeating the measurement, the mortar is remixed for two minutes, at speed two. With this procedure, the mortar has the same resting time[3] as applies for the concrete from remixing to repeating a viscometric measurement. An overview of the **measuring procedure** is listed in the following items. As applies on Page 86, the variable t designate the time elapsed from the point of water addition.

Box 4: Measuring procedure for mortar batches (old).

1. $t = 10$ min: Rheological measurement with the viscometer.
2. $t = 34$ min: Remixing for 2 minutes in the Hobart mixer, at speed two.
3. $t = 40$ min: Rheological measurement with the viscometer.
4. $t = 64$ min: Remixing for 2 minutes in the Hobart mixer, at speed two.
5. $t = 70$ min: Rheological measurement with the viscometer.
6. $t = 94$ min: Remixing for 2 minutes in the Hobart mixer, at speed two.
7. $t = 100$ min: Rheological measurement with the viscometer.

The mix designs of the mortars are shown in Table 4.4. As before, the **HMW Na** polymer is always used. The reproducibility results are shown in Figures 5.5 and 5.6. The sequence of numbers shown in both figures, are the **coefficient of variation** of the measurements and the **average** $\pm$ **standard deviation**. The right illustration of Figure 5.5, demonstrates the measured torque versus rotational frequency for batch 2 at $w/c = 0.4$, while the left illustration demonstrates the results in the usual compact form, only showing the corresponding yield value τ_o and plastic viscosity μ. Figure 5.6 demonstrates reproducibility results that apply for $w/c = 0.5$ and $w/c = 0.6$.

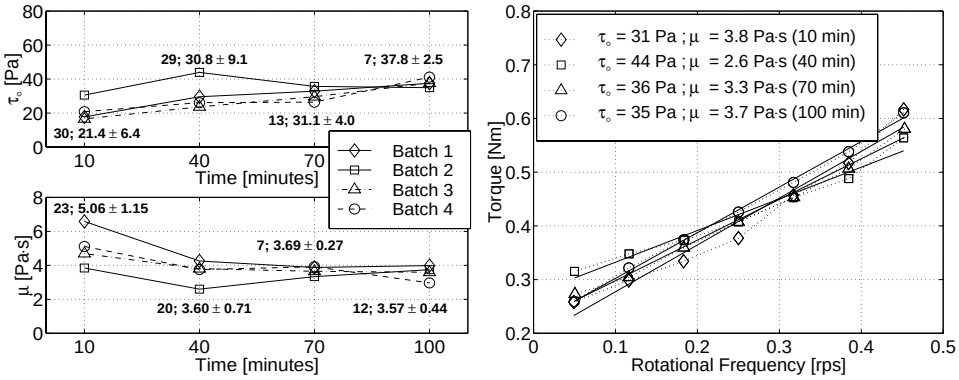


Figure 5.5: Reproducibility results at $w/c = 0.4$ for mortar using the mixing and measuring procedure of Box 3 and 4. To the left: Yield value τ_o and plastic viscosity μ from four repeated batches. To the right: The detailed result from the second batch. [OPC; $T_o = 23°C$].

The worst problem with the result shown in Figure 5.5 is perhaps not in terms of coefficient of variation (which are from about 5 to 30%), but rather in lack of time evolution in rheological values. As is explained in Section 6.4.1, it is to be expected that at least one of the rheological values τ_o or μ are increasing with time. As shown in the figure, neither τ_o or μ seems to have such properties; i.e. the 95% confidence interval

[3]Five minutes resting before the first measurement and then four minutes there after.

(in this case ±1.59·standard deviation, calculated according to Equation 6.2, Page 109 with $\nu = n - 1 = 3$ giving $t_{0.025} = 3.182$) completely overshadows any changes in viscometric values. This is perhaps more apparent with the right illustration, which shows how the torque measurements are more of the random and fluctuating nature (at least relative to what is shown in Figure 3.7, Page 61). This indicates that the measuring and mixing procedure of Box 3 and 4 are not really good.

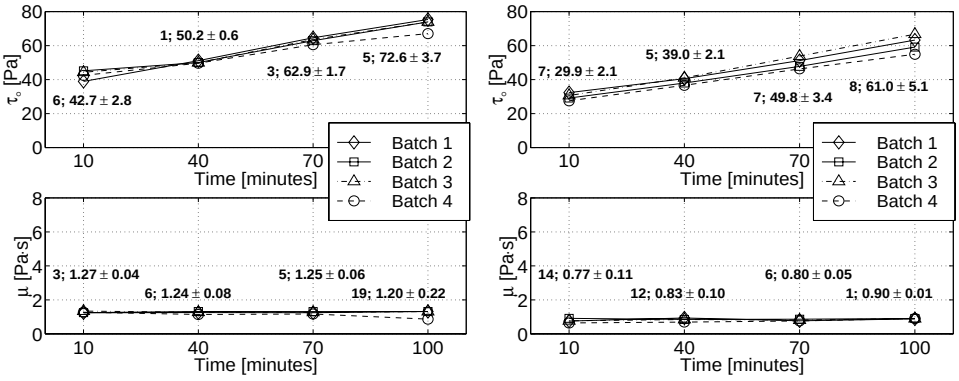


Figure 5.6: Reproducibility results at $w/c = 0.5$ (to the left) and at $w/c = 0.6$ (to the right) for mortar using the mixing and measuring procedure of Box 3 and 4. [OPC; $T_o = 23°C$].

The exact reason(s) for the strange results in the $w/c = 0.4$ batches is unclear. However, with the good reproducibility results for the $w/c = 0.5$ and $w/c = 0.6$ batch, in terms of smaller coefficient of variation (1 to 10%) and in terms of a clear yield value evolution with time, it can be suggested that the mixing procedure (see Box 3) is not intensive enough. This can be proposed in the light of that the m_a/m_c-ratio[4] is increasing from 1.4 to 2.0, when going from $w/c = 0.4$ to 0.6 (see Table 4.4), giving the basis for an increased effect of grinding and dispersion of the cement particles. Hence, the nominal same mixing procedure for the concrete and for the mortar at $w/c = 0.4$, is suspected not to be equal in the respect to grinding of the smaller cement clumps and in terms of dispersion of coagulated cement particles there after. As mentioned in Section 4.3, one of the reasons for using mortar rather than filler modified cement paste is to increase the grinding sufficiently, to be more similar to what applies inside the concrete.

5.3.3 Change in the Mixing and Measuring Procedure (OPC)

From the results shown in Figures 5.5 to 5.6, with increasing m_a/m_c-ratio (from $w/c = 0.4$ to $w/c = 0.6$), improved rheological results are produced. This is both in terms of coefficient of variation and in terms of a clear yield value τ_o evolution with time. However, the plastic viscosity μ remains unchanged as a function of time. With this information, a new mixing procedure was constructed of the more intensive type. An overview of this **new mixing procedure**, is shown in Box 5. The total mixing time, including dry mixing is now 10 minutes.

[4] m_a/m_c = mass ratio of aggregates versus cement. This ratio can give a measure of grinding and dispersing effect from the aggregates on the clumped and coagulated cement particles. A higher value means a higher effect.

As before, after each viscometric measurement, the mortar is placed back into the bowl of the Hobart mixer to rest. The resting time consist of less than 27 minutes, until remixing occurs. Care is taken to ensure a minimal loss of moisture (due to evaporation). Three minutes before commencing a viscometric measurement, the mortar is remixed for two minutes at speed 3. Therefore, the rest between a (re)mixing and a measurement is reduced down to 1 minute (from the previous 5 and 4 minutes). The modified **measuring procedure** is given in Box 6.

After each measurement, a crude cleaning procedure is applied, consisting of shaking the inner cylinder and removing the remaining mortar with cloth. Thereafter, the inner cylinder is covered with a plastic bag to inhibit a moisture loss from the remaining mortar, stuck on it. A plastic cover is also placed on top of the outer cylinder (i.e. the bucket of the viscometer). After two measurements, a more thorough cleaning procedure is applied, with the use of water (shown in Item 4, Box 6). This is introduced because a thin slam layers started to form on the inner cylinder, introducing potential error to the experimental setup.

Box 5: New mixing procedure for mortar batches.

1. Dry mixing of the $0-2\,\text{mm}$ aggregates and the cement at speed 1, in the time period of $t \in [-1\,\text{min}, 0\,\text{min}]$. This applies also for the old mixing procedure.
2. Wet mixing at speed 1, in the time period of $t \in [0\,\text{min}, 1\,\text{min}]$. The time $t = 0\,\text{min}$ is defined as the initial start of water addition. The water (which is premixed with lignosulfonate) is pored into the mix in the time interval of ca. 45 seconds. This applies also for the old mixing procedure.
3. Rest and hand mixing, in the time period of $t \in [1\,\text{min}, 4\,\text{min}]$, i.e. for 3 minutes. Previously, this period consisted of 2 minutes.
4. Mixing at speed 2, in the time period of $t \in [4\,\text{min}, 7\,\text{min}]$, i.e. for 3 minutes. Previously, this mixing was applied for 2 minutes.
5. Mixing at speed 3, in the time period of $t \in [7\,\text{min}, 9\,\text{min}]$, i.e. for 2 minutes. Relative to the old mixing procedure, this is a new addition.

Box 6: New measuring procedure for mortar batches.

1. $t = 10\,\text{min}$: Rheological measurement with the viscometer.
2. $t = 37\,\text{min}$: Remixing for 2 min at speed 3 in the Hobart mixer.
3. $t = 40\,\text{min}$: Rheological measurement with the viscometer.
4. Cleaning procedure applied to the inner and outer cylinder, with water.
5. $t = 67\,\text{min}$: Remixing for 2 min at speed 3 in the Hobart mixer.
6. $t = 70\,\text{min}$: Rheological measurement with the viscometer.
7. $t = 97\,\text{min}$: Remixing for 2 min at speed 3 in the Hobart mixer.
8. $t = 100\,\text{min}$: Rheological measurement with the viscometer.

With the new mixing and measuring procedure, the results shown in Figures 5.7 to 5.9 are produced. As shown in Figure 5.7, the reproducibility is much better compared to what is shown in Figure 5.5. This is both in terms of smaller coefficient of variation, and in terms of much clearer yield value evolution as a function of time.

The batch number 4 in Figure 5.7, demonstrates an unexpected low measured yield value τ_o throughout the whole time. This phenomenon recurred for batch number 1 at $w/c = 0.5$, shown in Figure 5.8. Before starting with the reproducibility series at

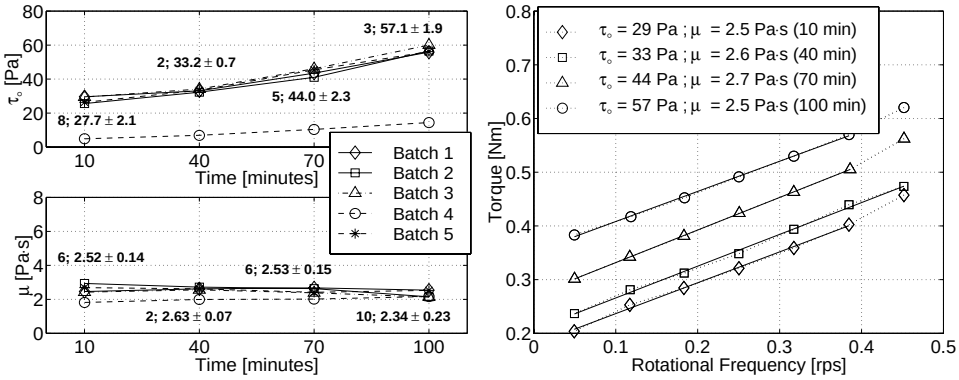


Figure 5.7: Reproducibility results at $w/c = 0.4$ for mortar using the mixing and measuring procedure of Box 5 and 6. To the left: Yield value τ_o and plastic viscosity μ from five repeated batches. To the right: The detailed result from the first batch. [OPC; $T_o = 23°C$].

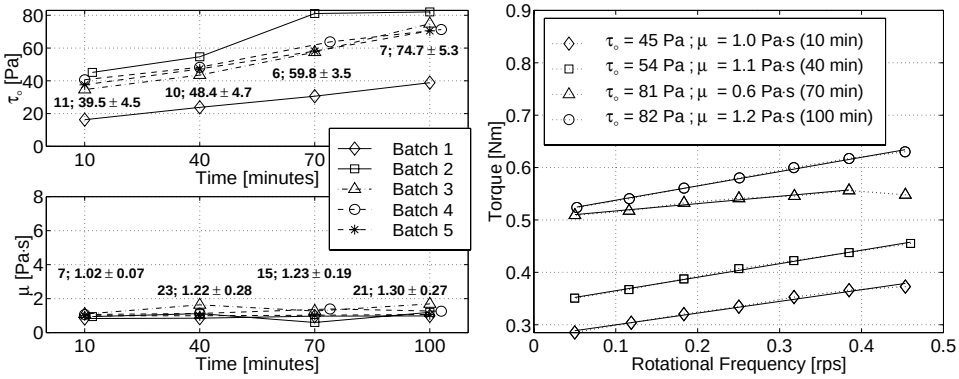


Figure 5.8: Reproducibility results at $w/c = 0.5$ for mortar using the mixing and measuring procedure of Box 5 and 6. To the left: Yield value τ_o and plastic viscosity μ from five repeated batches. To the right: The detailed result from the second batch. The first measurements occurred at 12 minutes after water addition. Previous two trials at 10 and 11 minutes gave a horizontal line, similar to the third measurements. [OPC; $T_o = 23°C$].

$w/c = 0.6$, the top ring of the ConTec Viscometer 4 is removed and the height h is instead manually measured for each measurement (see Figures 3.4 and 3.5, Page 55). The reason for the removal of the top ring, is due to the suspicion that some of the aggregates became stuck between the top ring and the inner cylinder, and hence inhibited the load cell from registering the correct torque. Such a phenomenon would lead to the same type of readings as shown for batch 4 at $w/c = 0.4$ and for batch 1 at $w/c = 0.5$.

Another problem, suspected to be related to the top ring, is demonstrated for the second batch at $w/c = 0.5$. This is shown in the right illustration of Figure 5.8: At $t = 70$ min, the measured yield value τ_o is somewhat higher than for the other batches at the same time. In fact, the same phenomenon occurred at $t = 10$ min (first trial) and at $t = 11$ min (second trial) for the same batch. With the third attempt at $t = 12$ min a reasonable result was finally gained. The exact reason for this phenomenon is somewhat unclear, but it is suspected that it originated from some

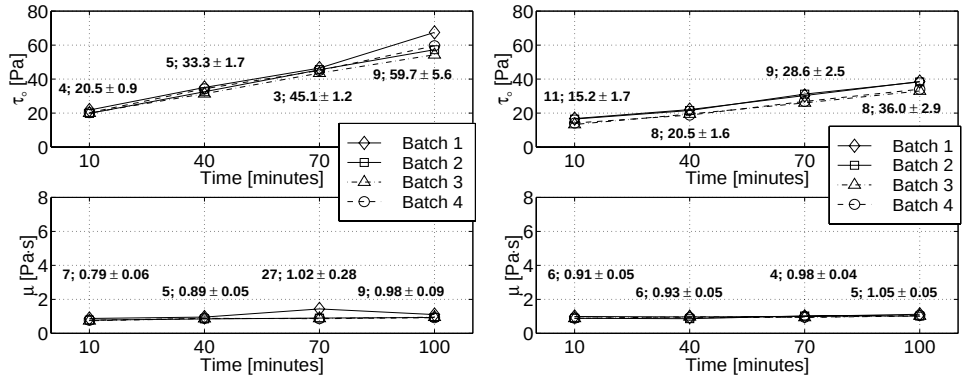


Figure 5.9: Reproducibility results at $w/c = 0.6$ for mortar using the mixing and measuring procedure of Box 5 and 6. [OPC; $T_o = 23°C$ (to the left); $T_o = 5°C$ (to the right)].

kind of interactions between the inner cylinder and the top ring, with the help of the $0 - 2$ mm aggregates. After removing the top ring, none of the problems of batch 4 at $w/c = 0.4$, batch 1 or 2 at $w/c = 0.5$ reoccurred for the whole experimental work of this thesis, and is therefore considered to be of a single incidence only. As a result of this, when calculating the coefficient of variation, average and standard deviation for the $w/c = 0.4$ and $w/c = 0.5$ batches, the abnormal values are not included.

The temperature change of each batch during the four measurements at 10, 40, 70 and 100 minutes, remains more or less a constant, with usually only minor variation of about $\pm 0.5°C$. However, a small variation is a reality from batch to batch, with a minimum value down to 21°C and maximum value of 24°C. However, the usual temperature was most often measured around 23°C.

5.3.4 Reproducibility at 5°C (OPC)

The mixing and measuring procedure are the same as used in Section 5.3.3 (Box 5 and 6). All the part materials, i.e. the cement, aggregates, water, plasticizers and the air-detraining agent, are stored in a special room (the "cold room") with the ambient temperature around $T_o = 5°C$. This was done at least 48 hours before mixing and rheological testing. Also, the outer cylinder of the ConTec Viscometer 4, the bowl and the whips of the Hobart mixer, are stored in this cold room, from the evening before. The mixing of the mortar took place with ambient temperature of 23°C. In that period, the bowl of the Hobart mixer is immersed in cooling-water at 5°C. Care was taken to have always equal amount of cooling-water for all the batches made.

After each viscometric measurement (except the last one), the mortar is placed back into the mixing bowl and stored in the cold room, for resting. The outer cylinder of the viscometer is also stored there, between measurements. Since remixing consisted of only two minutes, the cooling-water is not applied in such case.

The reproducibility results at $w/c = 0.6$ is shown in the right illustration of Figure 5.9. The $w/c = 0.4$ and $w/c = 0.5$ results are shown in Figure 5.10. For the $w/c = 0.4$ case, there is a peculiar increase and then a small decrease in yield value τ_o when using **HMW Na**. Such a profile is also produced when using the **HMW Ca** polymer, as is shown in Figure 6.1 (Page 104). To test if this phenomenon is related to the mixing procedure, additional measurement is commenced (batch 5), with a

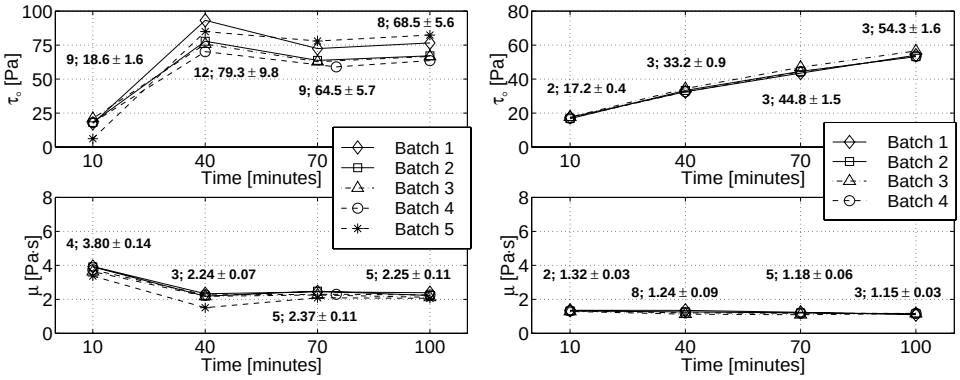


Figure 5.10: Reproducibility results at $w/c = 0.4$ (to the left) and at $w/c = 0.5$ (to the right) for mortar using the mixing and measuring procedure of Box 5 and 6. [OPC; $T_o = 5°C$].

more intensified mixing: The difference between this mixing procedure and the one described in Box 5, is that the mixing speed of 3 is used instead of 2, in the time period of $t \in [4\,\text{min}, 7\,\text{min}]$. Also, before the second measurement was made, the mortar was premixed for 4 minutes instead of 2 minutes (see Box 6). In spite of these changes, the peculiar evolution in yield value continued with the same trend.

The usual temperature for each batch at $t = 10\,\text{min}$ is around 9°C. With always a slight increase, the temperature at $t = 100\,\text{min}$ consist usually of about 10 to 11°C. The issue of temperature is considered further in Chapter 6.

5.3.5 Reproducibility at 38°C (OPC)

The mixing and measuring procedure are the same as used in Section 5.3.3 (Box 5 and 6). All the part materials, i.e. the cement, aggregates, water, plasticizers and the air-detraining agent, are stored in a special room (the "hot room") with the ambient temperature of $T_o = 38°C$. This was done at least 48 hours before mixing and rheological testing. Also, the HOBART mixer and the outer cylinder of the CONTEC VISCOMETER 4 are stored in the hot room, from the evening before. The mixing of the mortar took place inside this room, hence the ambient temperature condition during mixing is 38°C. The viscometer is stored in another room with ambient temperature of 23°C. The time period that the test material is exposed to the latter temperature, during each test and (de)mounting, is about 60 seconds.

The reproducibility results at 38°C is shown in Figure 5.11. The batch at $w/c = 0.6$ is excluded in the overall "38°C-series", hence the reproducibility tests are conducted only on mixes with $w/c = 0.4$ and $w/c = 0.5$.

The usual temperature for each batch at $t = 10\,\text{min}$ is around 38°C or just below. With always a decrease, the temperature at $t = 100\,\text{min}$ is 35 - 36°C. At least two factors contribute to this drop. First it is the heat convected from the test sample to the surroundings $\mathbf{q}_N$ (see Appendix B.7.2) when exposed to colder environment during a viscometric measurement. Secondly, due to the continuous evaporation of water, that costs heat (c.f. the *heat of vaporization* [154]), a drop in the temperature results. The same considerations apply for the $T_o = 5°C$ and $T_o = 23°C$-case, in Sections 5.3.4 and 5.3.3, respectively.

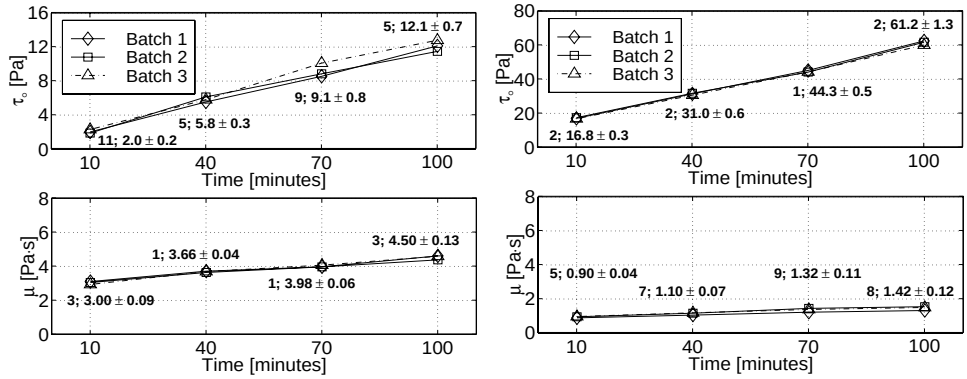


Figure 5.11: Reproducibility results at $w/c = 0.4$ (to the left) and at $w/c = 0.5$ (to the right) for mortar using the mixing and measuring procedure of Box 5 and 6. [OPC; $T_o = 38°C$].

5.4 Mortar Measurements (FAC)

The mixing and measuring procedure are the same as applies in Section 5.3.3 (Box 5 and 6). When beginning with the measurements for the $T_o = 23°C$-case, the similar mix design to Table 4.4 is used, with the same amount of binder as a constraint. The resulting mix design is shown in Table 4.5. Figure 5.12 demonstrates the results for the batch of $w/c = 0.4$. The coefficient of variation is roughly between 15 and 20%, which is rather high relative to the results shown in Figure 5.7 (roughly 2 to 5%).

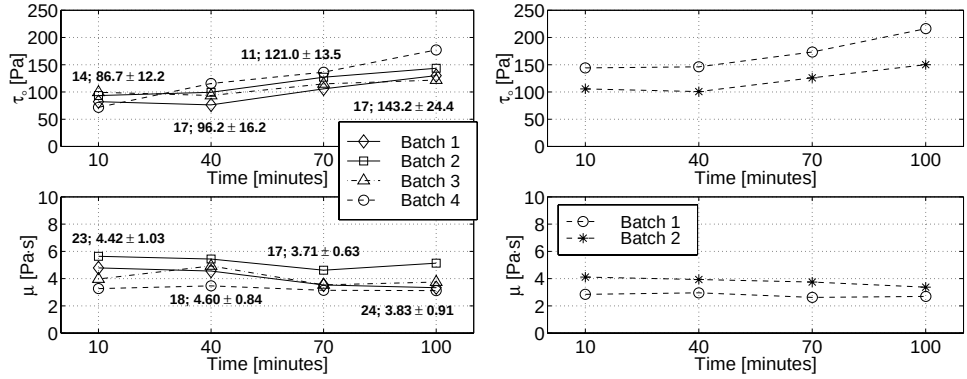


Figure 5.12: Reproducibility results at $w/c = 0.4$ for mortar using the fly ash cement (FAC). To the left: The mixing and measuring procedure of Box 5 and 6 is applied. To the right: A more intensified mixing is applied. [FAC; $T_o = 23°C$; Mix design is of Table 4.5].

The exact reason(s) for the lack of reproducibility in Figure 5.12 is unclear. However, in the light of the experience from Sections 5.3.2 and 5.3.3, a first attempt in compensating for this error, is to intensify the mixing further, relative to what is shown in Box 5 and 6. All mixing using speed 2 is increased up to speed 3 in the two boxes. Only two batches are tested with this modification. The corresponding rheological results are shown in the right illustration of Figure 5.12. As demonstrated, no advantage seems to results from this maneuver.

Changing the mixing and measuring procedure back to its original state of Box 5

and 6, and rather increasing the ratio of aggregates versus cement (m_a/m_c-ratio), gives the outcome shown in the left illustration of Figure 5.13. In both illustrations of this figure, the mix design is as shown in Table 4.6. As shown, the results improves considerably by this modification; i.e. the coefficient of variation is decreased from $11-24\%$ down to $2-9\%$. Same type of improvement is observed in Figures 5.5 and 5.6, as the m_a/m_c-ratio is increased from 1.4 to 2.0 (from $w/c = 0.4$ to $w/c = 0.6$). As suggested there, with increased m_a/m_c-ratio, a better grinding and dispersion occurs for the cement particles.

With the mix design of Table 4.6, the m_a/m_c-ratio is increased from 1.4 to 2.7. After changing the mix design in this manner, a different bath of sand is used. The grading of this sand is shown in Figure 4.4 and is marked as "$0-2$ mm (FAC)".

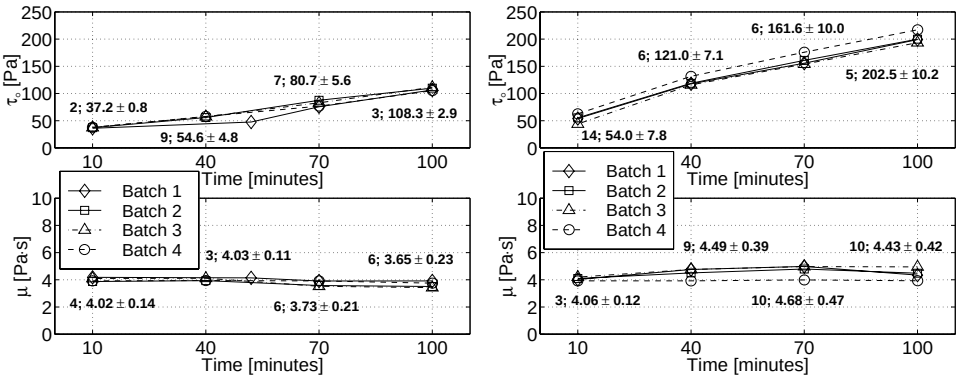


Figure 5.13: Reproducibility results at $w/c = 0.57$ for mortar using the mixing and measuring procedure of Box 5 and 6. [FAC; $T_o = 23°C$ (to the left); $T_o = 38°C$ (to the right); 0.6% sbwc **HMW Na**; Mix design is of Table 4.6]. For the results shown in this figure as well in all the previous figures, the **HMW Na** polymer is only used, c.f. the discussion in Section 5.2.2.

In the right illustration of Figure 5.13, reproducibility results are shown when the ambient temperature is $T_o = 38°C$. The temperature evolution of the test sample in the case of $T_o = 23°C$ and of $T_o = 38°C$ is as reported in the end of Sections 5.3.3 and 5.3.5, respectively.

5.5 Air Stability

When solving one problem, one is often creating a new one in the process. Unfortunately, there are no exceptions here. As mentioned previously, the reproducibility problems of mortars is solved by intensifying mixing, shown with Box 5 and 6. By doing this, air entrainment into the mortar is increased. Since air can have a major influence on rheological properties, the concern of air stability is a subject with its own section, presented here.

When doing an air measurement on mortar, a separate batch must be made. This is due to the limited volume capacity of the HOBART mixer. Furthermore, there is only sufficient material in each batch to make two air measurements. One measurement is made at 10 minutes and the other at 100 minutes, after water addition. However, the same mixing and measuring procedure is always applied as when doing a viscometric measurement (Box 5 and 6). That is, remixing is also applied at 40 and 70 minutes although no air measurements are conducted.

5.5.1 Air Stability of Concrete

Only one measurement of air content is made for each concrete batch, when doing a viscometric measurement. The air content is measured between the first and the second viscometric measurement. The average air content for all concrete batches is 2.1% with the standard deviation of 0.33%. More specifically, then for the concrete batches at the w/c-ratio of 0.4, 0.5 and 0.6, the air content is $1.9\% \pm 0.27\%$, $2.3\% \pm 0.30\%$ and $2.2\% \pm 0.17\%$, respectively (average±standard deviation). To investigate the air stability of concrete mixes as a function of time, additional batches are mixed. The results of this is shown in Table 5.1.

Table 5.1: Air measurements on concrete ($T_o = 23°C$).

Polymer	w/c	10 min	40 min	70 min	100 min	Average
HMW Na	0.4	2.3%	2.0%	2.3%	2.2%	2.2%
VHMW Na	0.4	0.7%	–	–	2.0%	1.4%
VHMW Na	0.5	1.5%	–	–	2.0%	1.8%
LMW Ca	0.5	2.3%	2.0%	2.2%	2.2%	2.2%
VHMW Na	0.6	2.2%	2.2%	2.2%	2.4%	2.3%

Except for when using **VHMW Na** at $w/c = 0.4$, the air contents of all mixes are relatively stable. For the **VHMW Na**-batch, the yield value at $t = 10$ min is lowest of all concrete measurements in this thesis. This is shown in Figure 6.1 (Page 104).

5.5.2 Air Stability of Mortar

Figure 5.14 shows the measured air content versus viscometric values τ_o and μ for mortar, at $w/c = 0.4$ (OPC). Each point in this figure, is extracted from two different measurements of two different batches of the same mix design. One is the air measurement and the other is the viscometric measurement. Relative to Table 5.1 for concrete, there is a much larger difference in air content present for this mortar case.

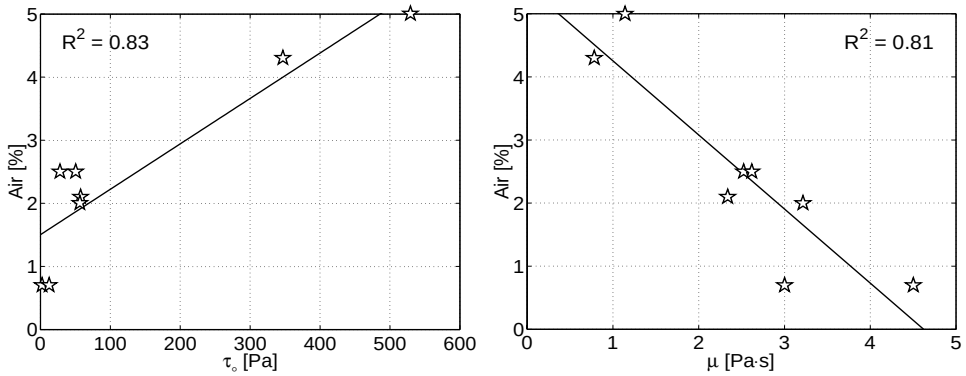


Figure 5.14: Air content plotted as a function of the yield value τ_o (to the left) and plastic viscosity μ (to the right). [**HMW Na** ∧ **LMWFS Ca**; 0.6% sbwc; $T_o = 23°C \wedge T_o = 38°C$; $w/c = 0.4$; OPC].

As shown with the left illustration of Figure 5.14, there is a strong relationship between the air content and the yield value τ_o ($R^2 = 0.83$), at $w/c = 0.4$ (OPC).

More precisely, with increasing yield value τ_o, the air content increases (and equally, vice versa). The right illustration shows also a strong relationship between air content and plastic viscosity μ ($R^2 = 0.81$), where with increasing plastic viscosity μ, the air content decreases (and vise versa).

Figure 5.15 shows a plot of air content versus viscometric values τ_o and μ, at $w/c = 0.5$ (OPC). As shown there, the relationship between air content and the viscometric values is not as clear as in Figure 5.14; i.e. there seems to be no relationship between viscometric values and air content at all. The coefficient of determination is $R^2 = 0.22$ in the left illustration, and $R^2 = 0.00$ in the right one (R is referred as the correlation coefficient [148]). Hence, there seems to be a good air stability present for the case of [$w/c = 0.5$; OPC]. On the other hand, for the FAC-case at $w/c = 0.57$ (see Figure 5.16), there is a stronger relationship between the air content and yield value τ_o, than initially shown in Figure 5.14. The coefficient of determination is large as $R^2 = 0.91$. However, the correlation is much lower between the air content and plastic viscosity μ ($R^2 = 0.40$) as shown in the right illustration of Figure 5.16.

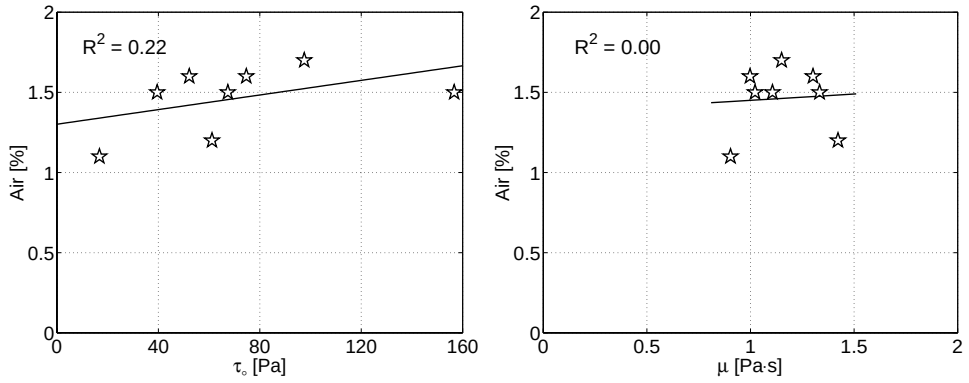


Figure 5.15: Air content plotted as a function of the yield value τ_o (to the left) and plastic viscosity μ (to the right). [**HMW Na** $\wedge$ **LMWFS Ca**; 0.3% sbwc; $T_o = 23°C \wedge T_o = 38°C$; $w/c = 0.5$; OPC].

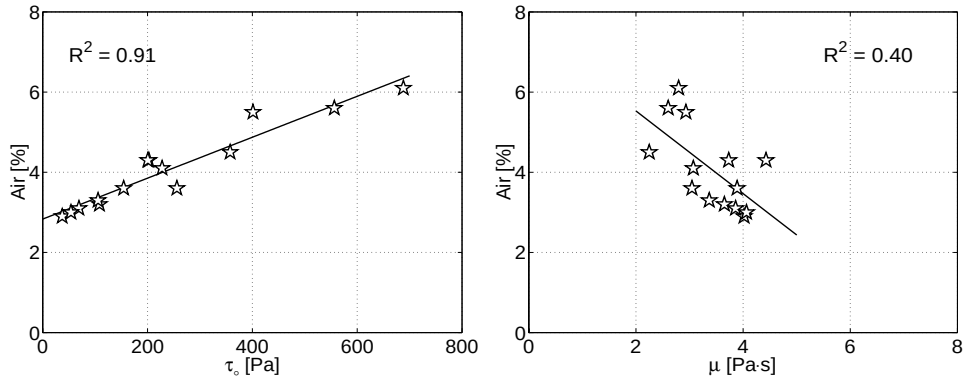


Figure 5.16: Air content plotted as a function of the yield value τ_o (to the left) and plastic viscosity μ (to the right). [**HMW Na** $\wedge$ **LMWFS Ca**; 0.6% sbwc $\wedge$ 0.3% sbwc; $T_o = 23°C$ $\wedge$ $T_o = 38°C$; $w/c = 0.57$; FAC].

With an air content of about 2% for the concrete, it is desirable to have an air content of about[5] 3.5% for the mortar. As shown in Figure 5.14, the measured air content is below and above this value. For the mortar case of $w/c = 0.5$ (Figure 5.15), the measured air content is somewhat lower than the target value. This could not be adjusted by decreasing the air-detraining agent (TBEP), because the unwanted air instability became then very much present.

5.5.3 Discussion and Conclusion

From Section 5.5.1, it is clear that the air content in concrete is not significantly changing from batch to batch or with time. Hence, air stability for the case of concrete is not a problem; i.e. τ_o and μ development is not caused by chancing air contents. Actually, the same applies for the mortar case of [w/c = 0.5 ; OPC] as shown in Figure 5.15. However, as shown in Figures 5.14 and 5.16, there is a certain relationship between the measured viscometric values τ_o and μ at the one hand, and the air content at the other. In relation to this, the immediate question is: *How much does the air content influence the viscometric values?*

Since the air content in concrete is always around 2% in the present work, it is safe to state that the change in viscometric values τ_o and μ, is not influenced by the air content. However, as is apparent from the previous figures, for some of the mortar batches, the air content is changing at the same time as other material parameters, like of the polymer dosage and polymer type used. This makes it difficult to separate the rheological effect of air content from the rheological effect of polymer type and polymer dosage (the two last effects are of primary interest in this thesis). Since there are no "air problems" for the concrete, it is interesting to plot the viscometric results of concrete (with complete air stability) versus the viscometric results of the corresponding mortar (with poor air stability). This is done in Figure 5.17. When considering this figure, one has to bear in mind that the error generated by plug flow for mortar (Section 3.5.3) and the error generated by the gravel migration for concrete (Chapter 10) are to some degree responsible for the scattering of the data shown.

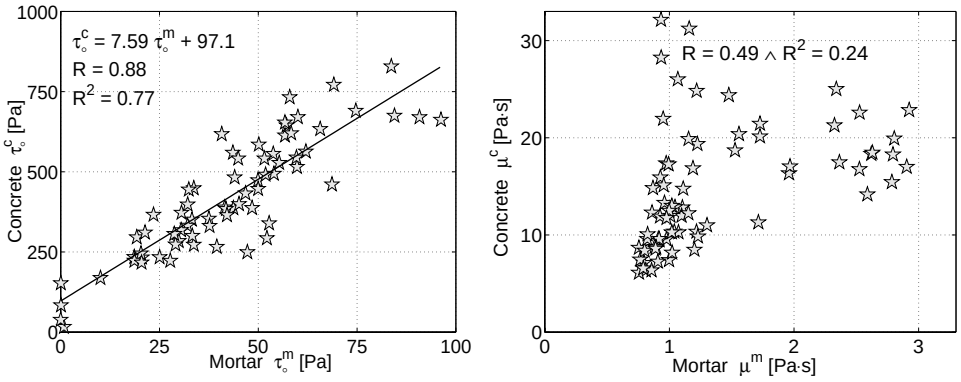


Figure 5.17: Relationship between measured viscometric value of concrete and mortar.

As shown in the left illustration of Figure 5.17, there is a certain relationship between the yield value τ_o of concrete and mortar. The correlation coefficient is

[5] For w/c-ratio at 0.4 and 0.5, the air content for the corresponding mortar should be 201/5741 = 3.48% and 201/5641 = 3.55%, respectively (values extracted from Table 4.3, Page 78).

$R = 0.88$ ($R^2 = 0.77$). With this relationship, it appears that the yield value τ_o of mortar is not much influenced by the change in air content. This is because of how the yield value behavior of mortar (with little air stability) can more or less imitate the yield value behavior of concrete (with complete air stability). However, for the plastic viscosity μ (the right illustration), the data is much more scattered ($R^2 = 0.24$), making it hard to ignore the possibility that the air content is changing the plastic viscosity μ of the mortar batches. Generally, for the fresh concrete, it has been demonstrated [144, 146] that the air content (within a certain range) has a little influence on yield value τ_o, while the plastic viscosity μ is more affected (see the illustration on Page 6). More precisely, with increased air content, the yield value τ_o remains more or less a constant, while the plastic viscosity μ is reduced [144, 146]. This result is not in a conflict with the above text.

Conclusion

It is not unlikely that different air content is somewhat disturbing the measured yield value τ_o of mortar. However, as is indicated from the above paragraph, this disturbance seems not to be large. Unfortunately, the same does not apply for the measured plastic viscosity μ. Rather, from Figures 5.14 and 5.16 it appears that with increased air content, the plastic viscosity μ decreases (as applies for the concrete case c.f. [144, 146]). The percentage error present in the measured plastic viscosity μ because of this, is unknown. This means that care must be taken when comparing plastic viscosity of two or more mortar batches. However, as is apparent in Figure 5.15 (Page 99), there exists a strong air stability for the case of [$w/c = 0.5$; OPC] in mortar, at least in the yield value range of about $\tau_o \in [0, 160]$ Pa (and for a moderate increase in yield value as a function of time). Therefore, for this case, the plastic viscosity evolution μ can be considered reliable, when considering the effect of different polymer type used. There are two exceptions for this, namely the **SNF**-case and the Without P/SP-case, at the temperature of 38°C. These two cases are discussed on Page 113.

A discussion about the errors caused by plug flow (Section 3.5.3) and by air entrainment is made in the beginning of Section 6.4, before discussing the overall experimental results of mortars.

Chapter 6

Experimental Results and Discussion

6.1 Introduction

In Sections 3.3 and 3.4, it is described how the viscometric values are retrieved from a single measurement. With the approach described, it is assumed that the test material is of a Bingham fluid (defined in Section 3.2). When considering the mortar, this assumption is acceptable as shown with the examples in Figures 5.7 and 5.8 (Page 93). However, at first consideration this assumption is harder to justify for the fresh concrete because of an additional flow phenomenon that is also present during a measurement. Realizing the nature of this additional flow phenomenon (done in Chapter 10), the Bingham model seems to apply to the concrete as well. The Bingham parameters for concrete are extracted in accordance with Section 3.4.2. The mixing and measuring procedure used, to characterize the rheological behavior of each batch as a function of time, is described in Chapter 5.

In Section 4.5, the test program of this thesis is presented. Its purpose is described in Section 1.2.1 and basically is to investigate how the effect of the different lignosulfonates changes as a function of temperature and time. In the current chapter (i.e. Chapter 6), the rheological results of this test program is given. This chapter is basically divided into two parts. First, in Section 6.2, an overview of all results is presented. Thereafter, a more detailed presentation and discussion are made of some interesting results of the first part.

6.2 Overview of Results

6.2.1 0.6% sbwc; w/c = 0.4; OPC

Yield Value τ_o

Figure 6.1 shows the measured yield value τ_o for all batches at $w/c = 0.4$ [0.6% sbwc; OPC]. The top center illustration is designated as "Concrete $T_o = 23°C$" and demonstrates the results of the concrete measurements with temperature of 23°C. See Section 5.2 for further readings about the concrete measurements. The three bottom figures, designated with "Mortar $T_o = 5°C$", "Mortar $T_o = 23°C$" and

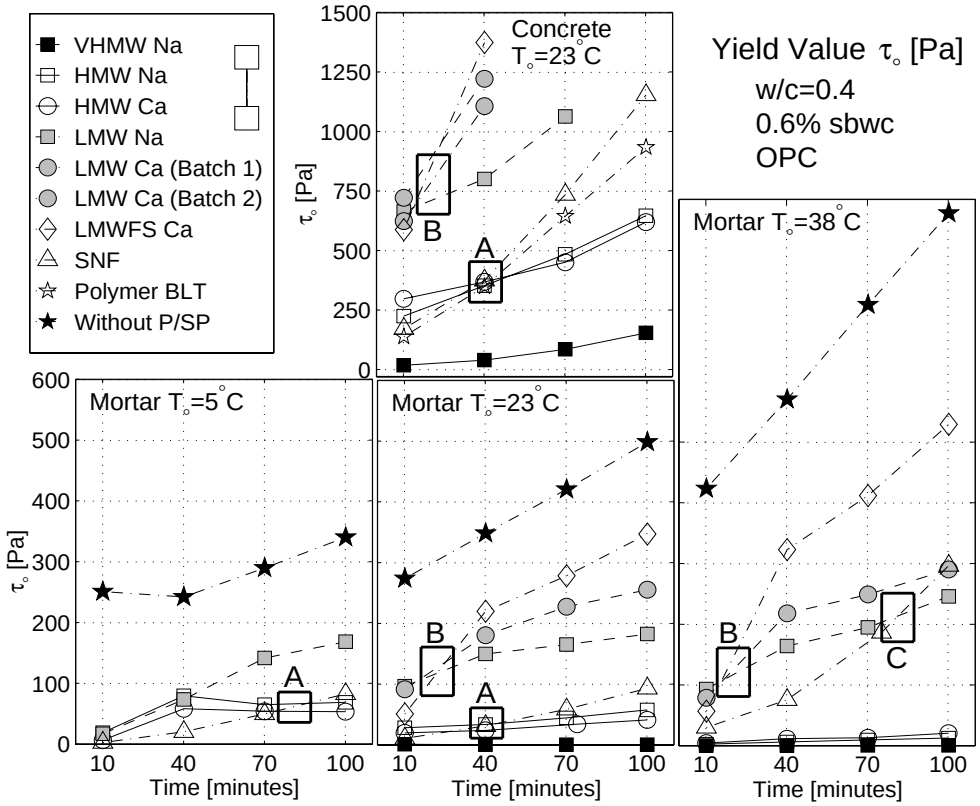


Figure 6.1: Measured yield value τ_o of concrete and mortar, using the different types of polymers ($w/c = 0.4$). The same results are shown in Figure 6.2 with error-bars and without markings. The corresponding result of plastic viscosity μ is shown in Figure 6.3.

"Mortar T_o = 38°C" demonstrates the results of the mortar measurements at the corresponding temperature of 5°C, 23°C and 38°C. For further readings about the temperature, see Section 5.3.4 for the $T_o = 5$°C-case, Section 5.3.3 for the $T_o = 23$°C-case and Section 5.3.5 for the $T_o = 38$°C-case.

Figure 6.2 demonstrates the same results as Figure 6.1 does, however with error-bars (namely $\pm\Delta\tau_o$) and without markings. The error-bar $\pm\Delta\tau_o$ of each measurement represents 95% confidence interval based on t-distribution and is calculated according to Equation 6.1 (Page 109). Also, in each illustration of Figure 6.2, there are number of four thick vertical lines shown ($\pm\Delta\check{\tau}_o$), connected with a horizontal line and ordered in a chronological order from left to right, applying at 10, 40, 70 and 100 minutes after water addition. These vertical lines of $\pm\Delta\check{\tau}_o$ represents 95% confidence interval generated from the reproducibility tests (at each w/c-ratio and at each temperature) in Chapter 5, for mix using the **HMW Na** polymer (see also Equation 6.2).

As shown in Figure 6.2, the difference in yield value τ_o between the batches of different polymers, are clearly significant for the mortar case. The difference for the concrete case is somewhat less significant, however the same trend is created for the concrete case as well for the mortar case at $T_o = 23$°C.

Of the lignosulfonate products used in the concrete batches, it is interesting to note the groupings in measured yield value τ_o in Figure 6.1. There are three distinct

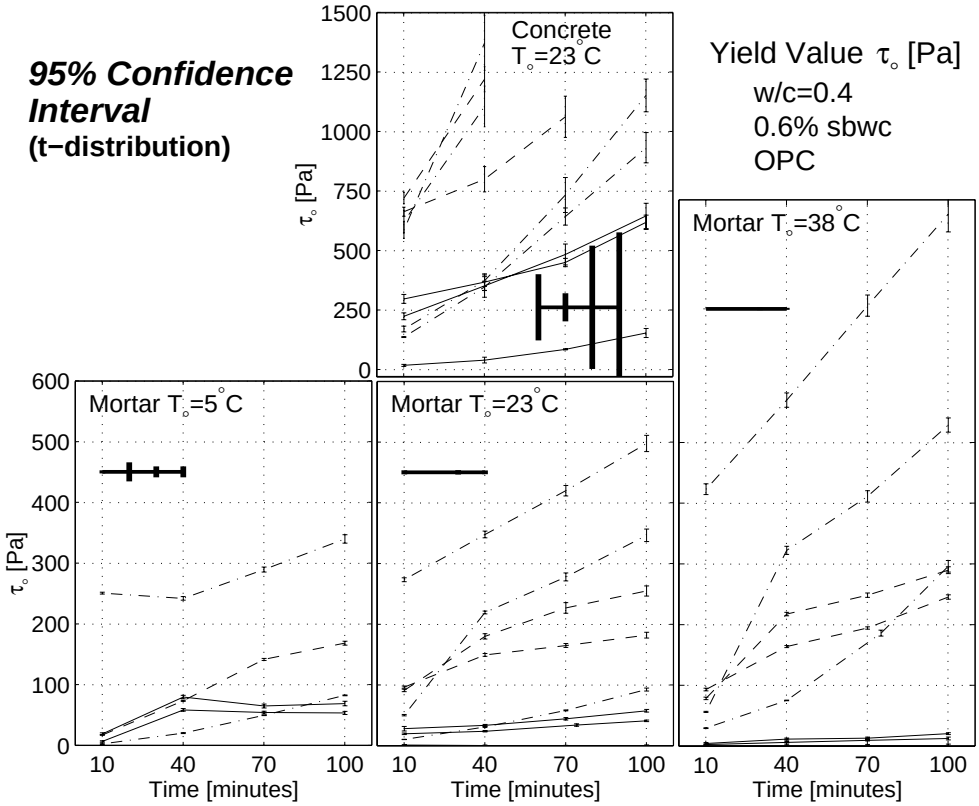


Figure 6.2: Measured yield value τ_o of concrete and mortar, using the different types of polymers ($w/c = 0.4$). The same results are shown in Figure 6.1 with markings and without error-bars. The corresponding result of plastic viscosity μ is shown in Figure 6.4.

different groups to be noted. The first one is the batch using the **VHMW Na** giving the lowest yield value measured. The second group consist of the batches using **HMW Na** and **HMW Ca**. The third group is of the batches using **LMW Na**, **LMW Ca** and **LMWFS Ca** polymers, giving the set of largest yield values measured. The characteristic difference between the polymers used, lies mainly in their molecular weight values M_n and M_m, shown in Table 4.2 (Page 74). Also, when comparing the concrete illustration and the 23°C-mortar illustration of Figure 6.1, the same rearrangement between the different batches is produced. This is shown for example with the crossing marked with the capital letters **A** and **B** in both illustration. Comparing the cases of $T_o = 23°C$ and $T_o = 38°C$ of mortar, the same rearrangement also exists, with the exception of the large temperature sensitivity of the batch using the **SNF** polymer. This is apparent with the crossing marked with the capital letter **C**. In the case of $T_o = 5°C$, some different rearrangement exists between the different batches, relative to the case of $T_o = 23°C$.

The product **Polymer BLT** in Figure 6.1 is not mentioned in Chapter 4 because it is applied only once. It is a 3rd generation of plasticizers, consisting of grafted carboxylic synthetic polymers. This product is not produced from lignin as the lignosulfonates are, and is therefore not a product of Borregaard LignoTech. The name used here is just a designation and does not present its commercial name. The

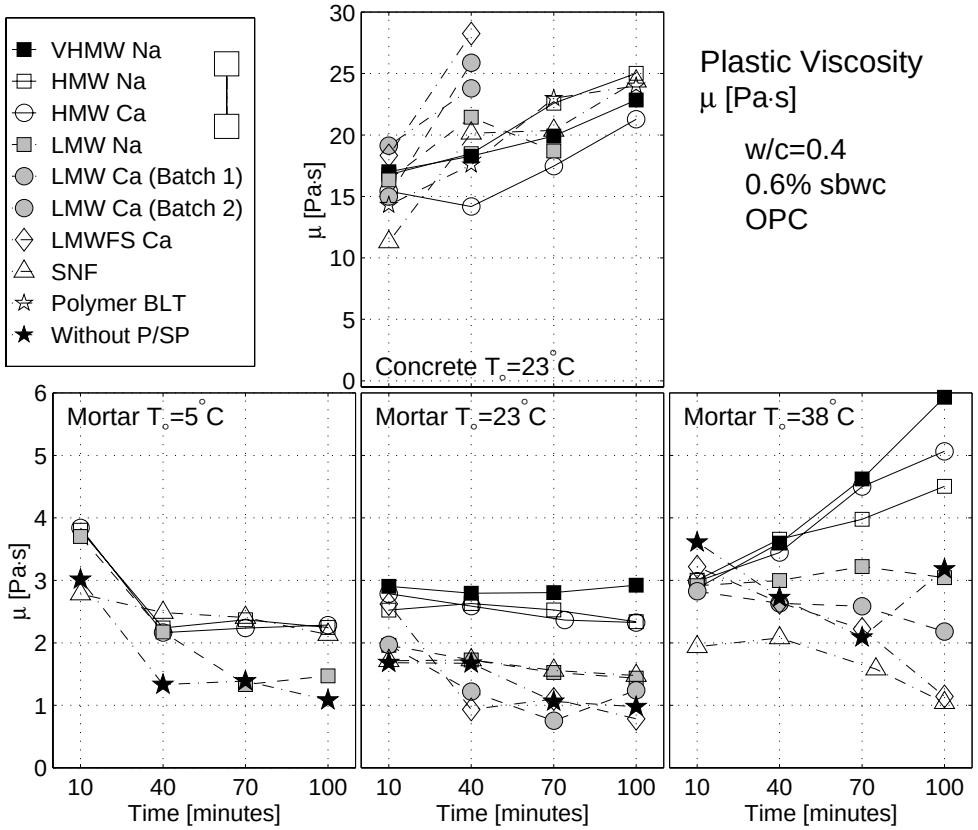


Figure 6.3: Measured plastic viscosity μ of concrete and mortar, using the different types of polymers ($w/c = 0.4$). The same results are shown in Figure 6.4 with error-bars and without markings. The corresponding result of yield value τ_o is shown in Figure 6.1.

amount of this polymer used, correspond to what was necessary to gain similar initial slump value to what applies for **HMW Na** and **HMW Ca**. This was done to be able to compare the workability loss between the polymers (see Figure 6.5). The amount of polymer used in this case is therefore 0.3% sbwc, instead of the usual 0.6% sbwc.

Plastic Viscosity μ

Figure 6.3 illustrates the measured plastic viscosity μ for all batches at $w/c = 0.4$ [0.6% sbwc; OPC]. Figure 6.4 shows the same results, however with the error-bars $\pm\Delta\mu$ and without markings. As before, the error-bar of each measurement represents 95% confidence interval, based on t-distribution and is calculated according to Equation 6.1. Also, in each illustration of Figure 6.4, there are number of four thick vertical lines shown ($\pm\Delta\breve{\mu}$), connected with a horizontal line and ordered in a chronological order from left to right and apply at 10, 40, 70 and 100 minutes. These vertical lines of $\pm\Delta\breve{\mu}$ represents 95% confidence interval generated from the reproducibility tests in Chapter 5 for the mix using the **HMW Na** polymer. The confidence interval $\pm\Delta\breve{\mu}$ is calculated according to Equation 6.2.

As shown in Figure 6.4, then for mortar at $T_o = 23°C$ and 38°C there is a difference in plastic viscosity μ for the batches using the **VHMW Na**, **HMW Na** and

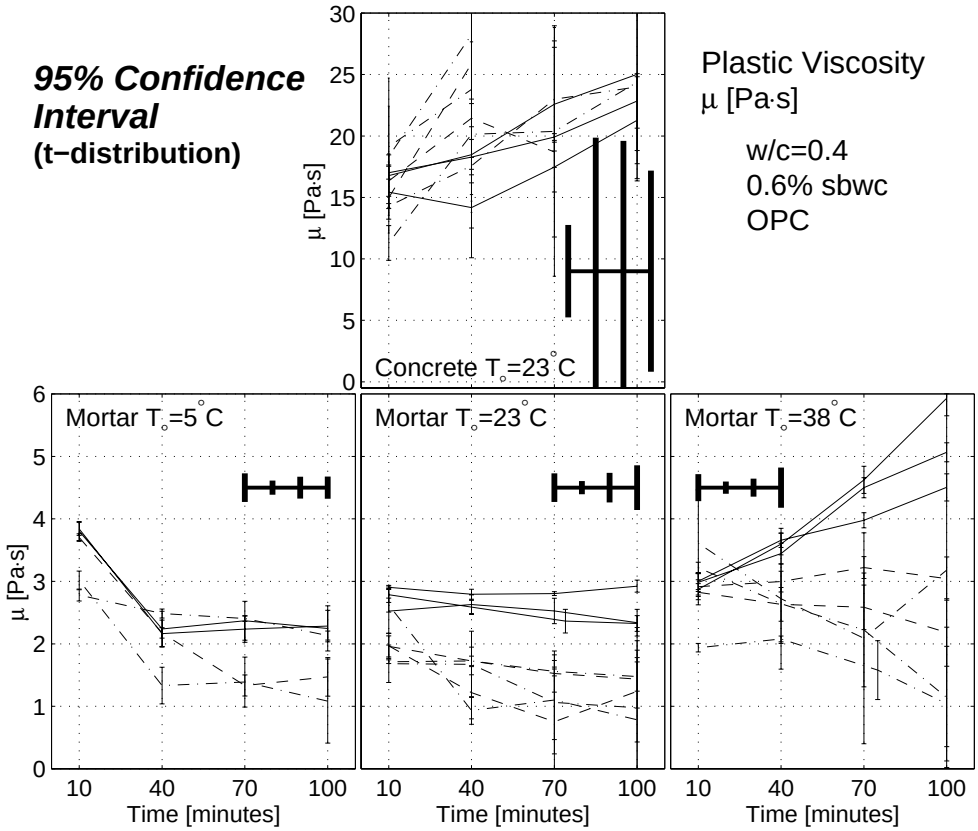


Figure 6.4: Measured plastic viscosity μ of concrete and mortar, using the different types of polymers ($w/c = 0.4$). The same results are shown in Figure 6.3 with markings and without error-bars. The corresponding result of yield value τ_o is shown in Figure 6.2.

HMW Ca at the one hand, and for the batches using the **LMW Na**, **LMW Ca** and **LMWFS Ca** at the other. However, this difference is somewhat reduced for the mortar case at $T_o = 5°C$. There is no significant difference in plastic viscosity μ between the different batches of concrete. Going back to Figure 6.3 (and ignoring confidence limits), similar rearrangement is produced for all mortar results and consist of that the batches using **VHMW Na**, **HMW Na** and **HMW Ca** polymers gives usually a larger plastic viscosity μ than the batches using **LMW Na**, **LMW Ca** and **LMWFS Ca** polymers. Note the combination of **high yield value** τ_o and **low plastic viscosity** μ that applies especially for the $T_o = 23°C$ and 38°C-cases of mortar. This type of combination is discussed in Section 6.2.3.

Slump Value *S*

Figure 6.5 demonstrates the measured slump value S for all batches at $w/c = 0.4$ [0.6% sbwc; OPC]. Due to the nature of the measurement, the calculation of confidence interval for a single slump measurement cannot be made. However, a 95% confidence interval $\pm\Delta S$ calculated from the reproducibility tests in Chapter 5 can be made. This is from the mix using the **HMW Na** polymer (see the left illustration of Figure 5.2). The calculation is done in the same manner as shown with Equation 6.2, namely with

$\Delta S = t_{0.025} P_{\text{std}}(S)/\sqrt{n}$. The result $\pm\Delta S$ is presented with four thick vertical lines, connected with a horizontal line. The lines are ordered in a chronological order from left to right and apply at 10, 40, 70 and 100 minutes. The lines extend over a relatively larger range, because of large $t_{0.025}$ value when $\nu = n-1 = 2$ (in Figure 5.2, the batch 4 is dismissed because of the uncontrolled increase in water content $\Rightarrow t_{0.025} = 4.303$).

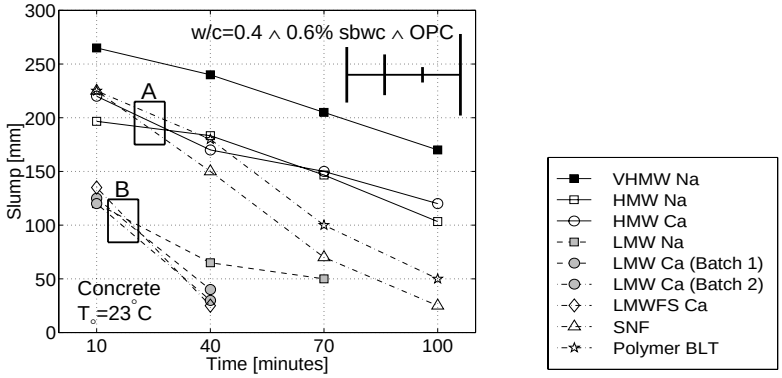


Figure 6.5: Measured slump value of concrete, using the different types of polymer.

If the above-mentioned 95% confidence interval where true for all the batches in Figure 6.5, there would not be a large significant difference between them. However, in this figure, the same rearrangement between the different batches is made as in Figure 6.1 (for both mortar and concrete), indicating the confidence limit shown in Figure 6.5 could be somewhat overestimated. This is shown for example with the crossing marked with the capital letters **A** and **B** in both figures. Of the lignosulfonate products used, there are three distinct different groups present in Figure 6.5 (as well in Figure 6.1). The first group is the batch using the **VHMW Na** giving the largest slump value measured. The second group consist of the batches using **HMW Na** and **HMW Ca**. The third group is of the batches using **LMW Na**, **LMW Ca** and **LMWFS Ca** polymers and are of the lowest slump values measured.

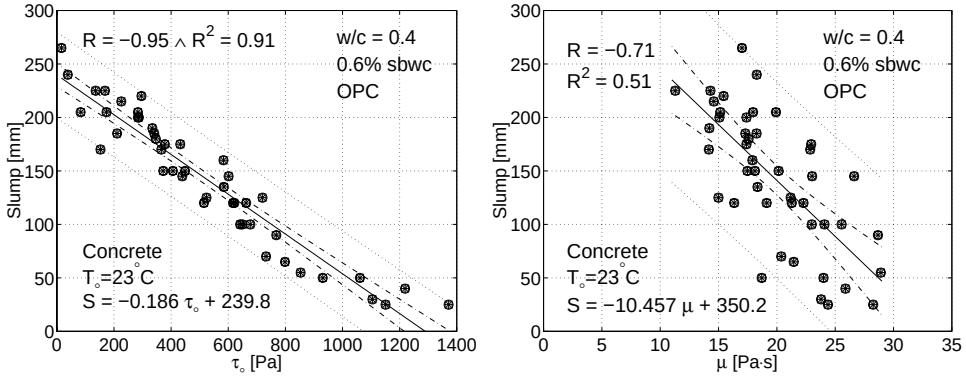


Figure 6.6: Comparison of measured slump value versus the measured yield value τ_o (to the left) and plastic viscosity μ (to the right). These results are also shwon in Figure 10.17 (Page 251).

With the good correspondence between the results in Figure 6.1 and in Figure 6.5, a good relationship between the slump value and the yield value τ_o is a reality. This

is shown in the left illustration of Figure 6.6. The right illustration demonstrates the somewhat weaker relationship between the slump value and the plastic viscosity μ. In both illustrations, the dashed dotted line represents the 95% confidence interval (based on t-distribution) for the regressed lines $S = b_1\,\tau_o + a_1$ and $S = b_2\,\mu + a_2$. That is, it is with 95% confidence that the true (and unknown) functions $S = S(\tau_o)$ and $S = S(\mu)$ exists within the boundary of the two dashed dotted lines of each illustration. The dotted lines represents the 95% prediction interval (based on t-distribution); i.e. there is a 95% probability that the next measurement falls within the boundary of the two dotted lines of each illustration. For further readings about this type of statistical approach, see the textbook by Walpole et al. [148].

6.2.2 Calculation of Confidence Interval

In Figures 6.2 and 6.4, viscometric results are shown with error-bars. Also, in each illustration there are number of four thick vertical lines shown, connected with a horizontal line and ordered in a chronological order from left to right and apply at 10, 40, 70 and 100 minutes after water addition. In this section, a description and a short demonstration of how these errors are calculated is given.

To calculate the 95% confidence interval of the slope H and the point of intersection with the ordinate G (see Equation 3.25, Page 59), one must first calculate their standard error [128] $\alpha_H = H\sqrt{((1-R^2)/R^2)(1/(n-1))}$ and $\alpha_G = \alpha_H\sqrt{\sum f_o^2/n}$. The term R is the correlation coefficient, n is the number of torque points used in each measurement and $\sum f_o^2$ is the square sum of the rotational frequency[1] applied. The 95% confidence interval of H and G is given as $\Delta H = t_{0.025}\,\alpha_H$ and $\Delta G = t_{0.025}\,\alpha_G$ respectively [128]. The term $t_{0.025}$ is the 95% confidence value of the t-distribution. Depending on the degree of freedom $\nu = n-2$, the term $t_{0.025}$ hold a different values. In the case of mortar with $n = 7$, the degree of freedom is $\nu = n-2 = 5$ giving $t_{0.025} = 2.571$. In the case of concrete with $n = 5$, the degree of freedom is $\nu = n-2 = 3$ resulting in $t_{0.025} = 3.182$. Using Equations 3.26 and 3.27 to normalize ΔH and ΔG, in creating $\Delta\mu$ and $\Delta\tau_o$, results in Equation 6.1.

$$\Delta\mu = \frac{\Delta H\,(R_i^{-2} - R_o^{-2})}{8\,\pi^2\,h} \quad \wedge \quad \Delta\tau_o = \frac{\Delta G\,(R_i^{-2} - R_o^{-2})}{4\,\pi\,h\,\ln(R_o/R_i)} \qquad (\nu = n-2) \quad (6.1)$$

Thus based on a single measurement, it is with 95% confidence that the true viscometric values are $\mu \pm \Delta\mu$ and $\tau_o \pm \Delta\tau_o$. The error-bar of each measurement shown in Figures 6.2 and 6.4, consist of $\pm\Delta\tau_o$ and $\pm\Delta\mu$ respectively.

In each illustration of Figures 6.2 and 6.4, there are number of four thick vertical lines shown, connected with a horizontal line. The lines are ordered in a chronological order from left to right and apply at 10, 40, 70 and 100 minutes. These lines are generated from the reproducibility tests in Chapter 5 and represents the calculated 95% confidence interval for the mix using the **HMW Na** polymer. The 95% confidence for the reproducibility tests are simply calculated with Equation 6.2 [148].

$$\Delta\breve{\mu} = \frac{t_{0.025}P_{std}(\mu)}{\sqrt{n}} \quad \wedge \quad \Delta\breve{\tau}_o = \frac{t_{0.025}P_{std}(\tau_o)}{\sqrt{n}} \qquad (\nu = n-1) \quad (6.2)$$

The term P_{std} is the standard deviation, shown for example in Figure 5.2 (Page 87). Thus based on measurements of several repeated batches, it is with 95% confidence

[1] For mortar: $\sum f_o^2 = 0.45^2+0.38^2+0.32^2+0.25^2+0.18^2+0.12^2+0.05^2 = 0.56\,s^{-2}$ (see Figure 3.6, Page 60). For concrete: $\sum f_o^2 = 0.37^2 + 0.30^2 + 0.23^2 + 0.16^2 + 0.09^2 = 0.31\,s^{-2}$ (see Figure 3.9).

that the true viscometric values are $\bar{\mu} \pm \Delta\breve{\mu}$ and $\bar{\tau}_o \pm \Delta\breve{\tau}_o$, where $\bar{\mu}$ and $\bar{\tau}_o$ are the average values. When presenting the result of the **HMW Na**-case, it is always the average values $\bar{\mu}$ and $\bar{\tau}_o$ that are shown. The values of $\pm\Delta\breve{\mu}$ and $\pm\Delta\breve{\tau}_o$ give a better presentation of the actual error, than the values of $\pm\Delta\mu$ and $\pm\Delta\tau_o$. This is because of how the former values also include the additional errors generated by the operator while weighing the sub-materials (water, cement and so forth) and in the mixing to create the corresponding batch to be tested. Although the same mixing procedure is always applied (see Box 5, Page 92), the mixing inside the Hobart mixer was observed to vary somewhat from batch to batch.

6.2.3 0.3% sbwc; w/c = 0.5; OPC

Yield Value τ_o

Figure 6.7 illustrates the measured yield value τ_o for all batches at $w/c = 0.5$ [0.3% sbwc; OPC]. Figure 6.8 shows the same results, however with the error-bars $\pm\Delta\tau_o$ (Equation 6.1) and the four vertical lines $\Delta\breve{\tau}_o$ (Equation 6.2), and without markings.

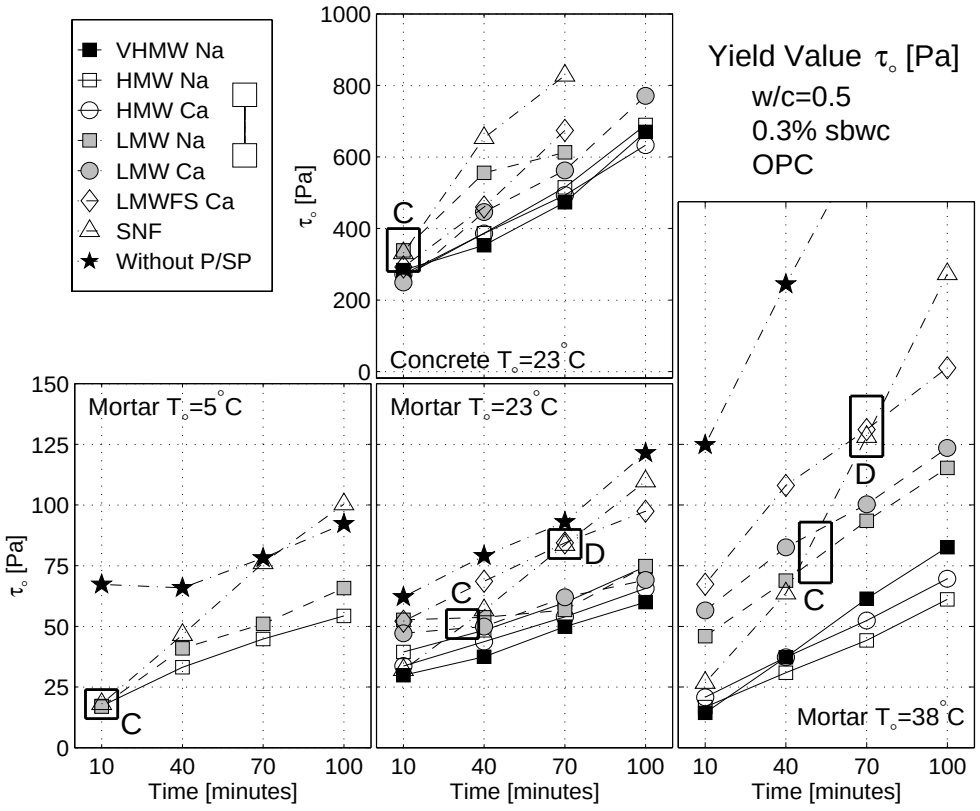


Figure 6.7: Measured yield value τ_o of concrete and mortar, using the different types of polymers ($w/c = 0.5$). The same results are shown in Figure 6.8 with error-bars and without markings. The corresponding result of plastic viscosity μ is shown in Figure 6.9.

As applies in Section 6.2.1, the error-bar $\pm\Delta\tau_o$ of each measurement in Figure 6.8 represents 95% confidence interval, calculated according to Equation 6.1. The four thick vertical lines $\pm\Delta\breve{\tau}_o$ also shown, represents the 95% confidence interval calculated

from the four (and sometimes, three) repeated batches using the **HMW Na** polymer (extracted from the reproducibility tests of Chapter 5). The $\Delta\breve{\tau}_o$ value is calculated according to Equation 6.2. For example, in the case of mortar at $T_o = 23°C$ (the bottom center illustration in Figure 6.8), these lines are calculated from the standard deviation shown in the top left illustration of Figure 5.8 (Page 93), with $\nu = n - 1 = 4 - 1 = 3$ degrees of freedom (there, the batch number 1 is excluded). In the case of concrete, the degree of freedom for the four vertical lines is $\nu = n - 1 = 3$, instead of the previous 2 that applies in Section 6.2.1. It is interesting to note in Figure 6.8 how the $\pm\Delta\breve{\tau}_o$ values are increasing with time from water addition. At 10 minutes, the values of $\pm\Delta\breve{\tau}_o$ extend over similar range as the value of $\pm\Delta\tau_o$. However at 100 minutes, the values of $\pm\Delta\breve{\tau}_o \approx \pm 200\,\text{Pa} \Rightarrow 400\,\text{Pa}$ extend almost over the half part of the scale used in this illustration.

As shown in Figure 6.8, the difference in yield values between the batches of different polymers, are clearly significant for the mortar case. The difference for the concrete case is some less significant, however roughly the same trend is created for the concrete case as well for the mortar case at $T_o = 23°C$.

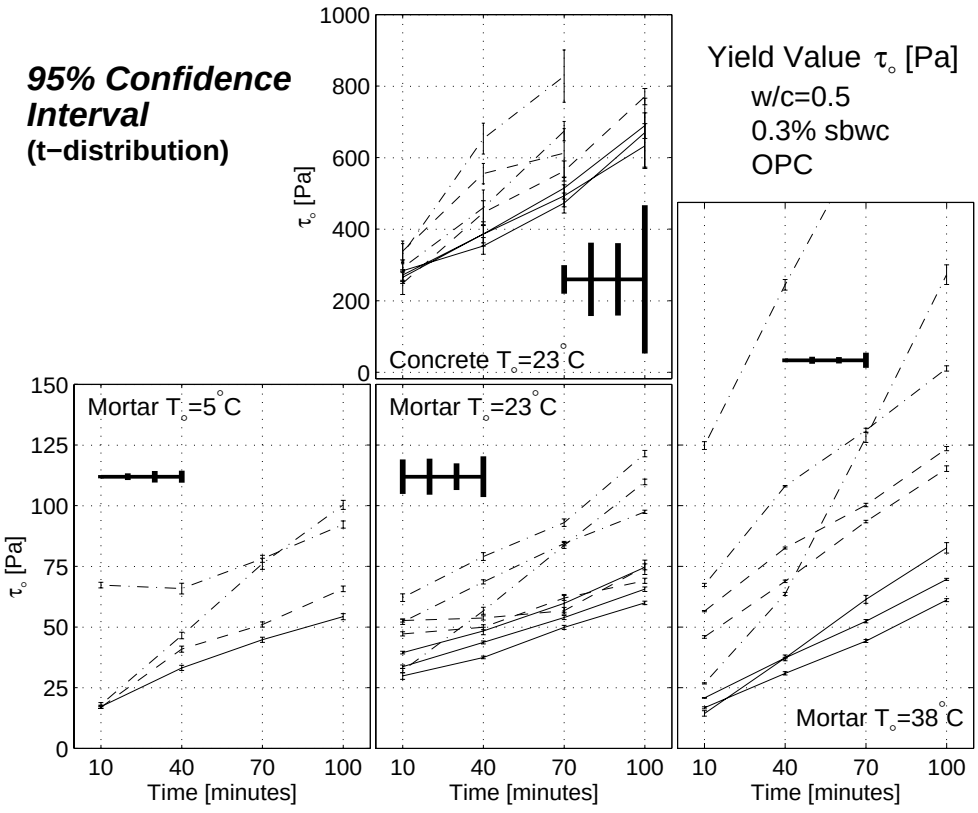


Figure 6.8: Measured yield value τ_o of concrete and mortar, using the different types of polymers ($w/c = 0.5$). The same results are shown in Figure 6.7 with markings and without error-bars. The corresponding result of plastic viscosity μ is shown in Figure 6.10.

When considering the overall results shown in Figure 6.7, there is a less difference between the different batches than applies in Figure 6.1. This is due to the less amount of plasticizing polymer used (0.6% → 0.3% sbwc) and the increased w/c-ratio

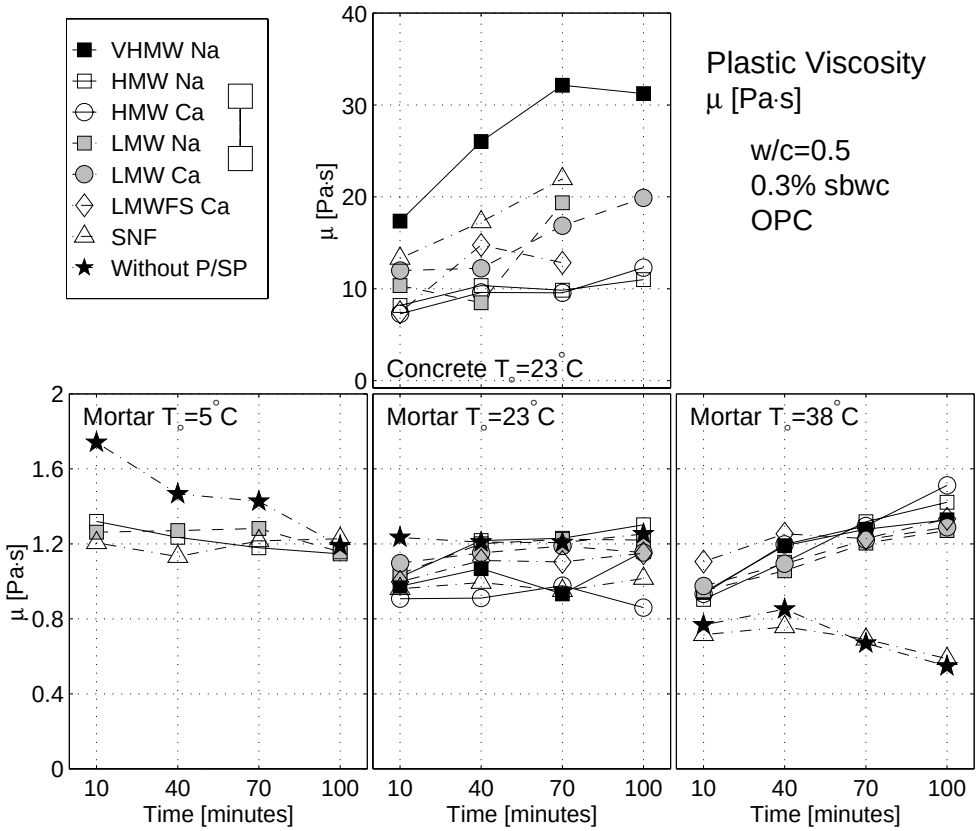


Figure 6.9: Measured plastic viscosity μ of concrete and mortar, using the different types of polymers ($w/c = 0.5$). The same results are shown in Figure 6.10 with error-bars and without markings. The corresponding result of yield value τ_o is shown in Figure 6.7.

used in each batch ($w/c = 0.4 \rightarrow 0.5$). Peculiarly, here the **VHMW Na** polymer performs equally as well as the **HMW Na** and **HMW Ca** polymers. This is not the case in Figure 6.1.

The general trend of all illustrations in Figure 6.7, is that the there are roughly two different groups. One is the batches using the **VHMW Na**, **HMW Na** and **HMW Ca** giving the lowest yield value measured. The second group consist of the batches using **LMW Na**, **LMW Ca** and **LMWFS Ca** polymers, giving the set of largest yield values measured. A familiar crossing between the **LMW Na**, **LMW Ca**-batches and the **SNF**-batch is measured and marked with the capital letter **C** in Figure 6.7. In Figure 6.1, this crossing is also marked with **C**.

Plastic Viscosity μ

Figure 6.9 demonstrates measured plastic viscosity μ for all batches at $w/c = 0.5$ [0.3% sbwc; OPC]. Figure 6.10 shows the same results, however with the error-bars $\pm\Delta\mu$ (Equation 6.1) and the four vertical lines $\Delta\breve{\mu}$ (Equation 6.2), and without markings.

As shown in Figure 6.10, there is not much significant difference between the measured plastic viscosity μ of the different batches, either for concrete or mortar. As such, there is not much basis for discussing these data. There is however one interest-

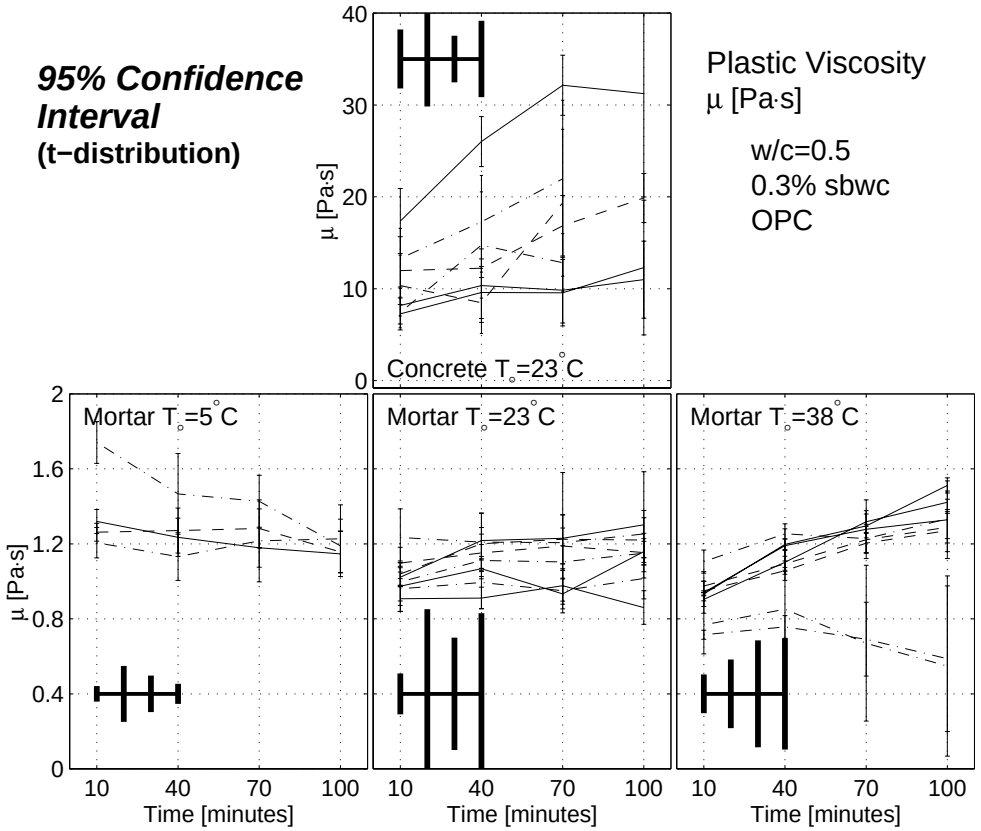


Figure 6.10: Measured plastic viscosity μ of concrete and mortar, using the different types of polymers ($w/c = 0.5$). The same results are shown in Figure 6.9 with markings and without error-bars. The corresponding result of yield value τ_o is shown in Figure 6.8.

ing phenomenon that occurs at 38°C, for the mortar cases of **SNF** and Without P/SP. Both of them have a significant lower plastic viscosity μ, relative to the other batches tested. That is, the batches of highest yield value τ_o (and of highest increase in yield value) are also of lowest plastic viscosity μ. This type of trend is also observed when comparing Figures 6.3 and 6.1 ($w/c = 0.4$), for the cases of **SNF**, Without P/SP and **LMWFS Na**, at 38°C. As mentioned in Sections 5.5.2 and 5.5.3, air entrainment has a minor effect on the yield value τ_o of mortar. Also, as shown in the left illustration of Figures 5.14 and 5.16 (Page 98), an increased yield value τ_o parallel an increased air content. A plausible explanation of this could be that with increased yield value τ_o, the buoyancy force of the entrained air (mostly entrained through mixing and remixing of Box 5 and 6, Page 92) is not sufficient to allow it to surface; i.e. viscous stresses are large enough to keep the air bubbles submerged. Hence, with increasing yield value, more air remains in the mortar batches. From Sections 5.5.2 and 5.5.3, it appears that with increasing air content, the plastic viscosity μ decreases (as applies for the concrete case c.f. [144, 146]). Hence, from the above text, one could consider the following sequence: Increased yield value τ_o as a function of time ⇒ Increased air content as a function of time ⇒ Reduced plastic viscosity μ as a function of time. If this sequence do apply, it would explain the combination of high yield value τ_o (and high increase in yield value) and low plastic viscosity μ, as previously observed.

Slump Value S

Figure 6.11 shows the measured slump value S for all batches of $w/c = 0.5$ [0.3% sbwc; OPC]. Due to the nature of the measurement, the calculation of confidence interval for a single slump measurement cannot be made. However, a 95% confidence interval calculated from the reproducibility tests in Chapter 5 can be made. The result is presented with four thick vertical lines, namely $\pm\Delta S = \pm t_{0.025} P_{\text{std}}(S)/\sqrt{n}$ (see Equation 6.2), connected with a horizontal line. The lines are ordered in a chronological order from left to right and apply at 10, 40, 70 and 100 minutes after water addition. These lines are generated from the mix using the **HMW Na** polymer (see the left illustration of Figure 5.3, Page 88).

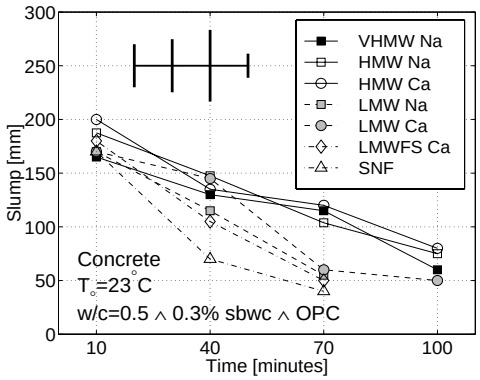


Figure 6.11: Measured slump value of concrete, using the different types of polymer.

In Figure 6.11, the same rearrangement between the different batches is made as for the case in Figure 6.7: There is a less variation between the different batches than applies in Figure 6.5. This is most likely due to the less amount of plasticizing polymer used (0.6% → 0.3% sbwc) and the increased w/c-ratio used in each batch ($w/c = 0.4 \rightarrow 0.5$). What is measured by the viscometer, is also observed with the slump values. For example, the **VHMW Na** polymer is shown to perform equally as well as the **HMW Na** and **HMW Ca** polymers, by both measurements.

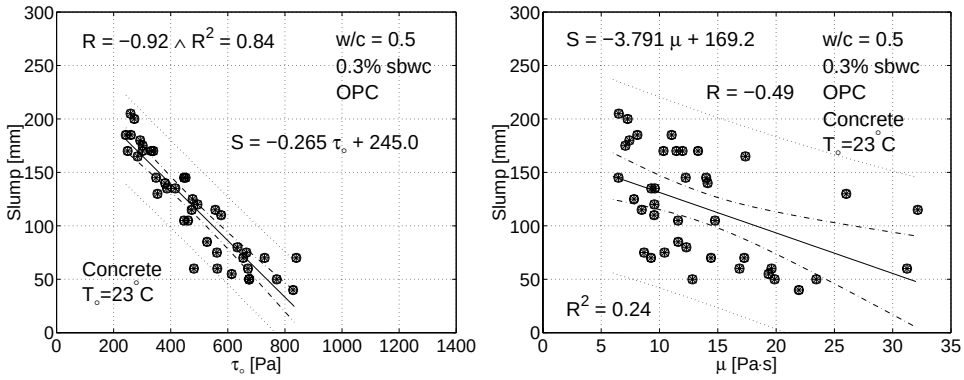


Figure 6.12: Comparison of measured slump value versus the measured yield value τ_o (to the left) and plastic viscosity μ (to the right).

With the good correspondence between the results in Figure 6.7 and in Figure 6.11, a good relationship between the slump value and the yield value τ_o is a reality. This is shown in the left illustration of Figure 6.12. The right illustration demonstrates the considerable weaker relationship between the slump value and the plastic viscosity μ. As discussed in Section 6.2.1, the dashed dotted line represents the 95% confidence interval (based on t-distribution) for the regressed lines $S = b_1\,\tau_o + a_1$ and $S = b_2\,\mu + a_2$. That is, it is with 95% confidence that the true (and unknown) functions $S = S(\tau_o)$ and $S = S(\mu)$ exists within the boundary of the two dashed dotted lines of each illustration. The dotted lines represents the 95% prediction interval (based on t-distribution); i.e. there is a 95% probability that the next measurement falls within the boundary of the two dotted lines of each illustration.

6.2.4 0.1% sbwc; w/c = 0.6; OPC

Yield Value τ_o and Plastic Viscosity μ

Figure 6.13 shows the measured yield value τ_o and plastic viscosity μ at $w/c = 0.6$ [0.1% sbwc; OPC]. Figure 6.14 illustrates the same results, however with the error-bars $\pm\Delta\tau_o$ (Equation 6.1) and the four vertical lines $\Delta\check{\tau}_o$ (Equation 6.2), and without markings (see Section 6.2.2).

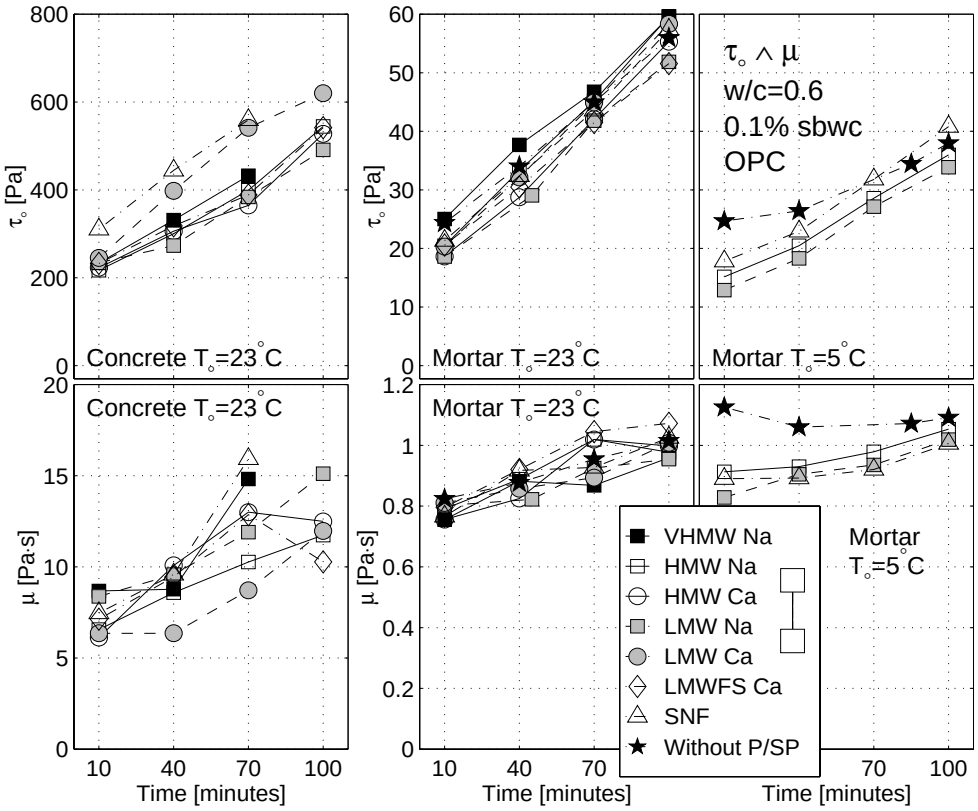


Figure 6.13: Measured yield value τ_o and plastic viscosity μ of concrete and mortar, using the different types of polymers ($w/c = 0.6$). The error-bars are shown in Figure 6.14.

As shown in Figure 6.14, the difference in yield value τ_o, or in plastic viscosity μ

between the batches of different polymers, are not significant. As such, there is no basis for discussing the effects of the polymers used for the overall mix of $w/c = 0.6$. Because of the insignificant difference between the different batches, the experiment with the temperature of $T_o = 38°C$ was not conducted.

As applies in Section 6.2.1, the error-bar $\pm\Delta\tau_o$ of each measurement in Figure 6.14 represents 95% confidence interval, calculated according to Equation 6.1. The four thick vertical lines $\pm\Delta\breve{\tau}_o$ also shown, represents the 95% confidence interval calculated from the four repeated batches using the **HMW Na** polymer (extracted from the reproducibility tests of Chapter 5). The $\Delta\breve{\tau}_o$ value is calculated according to Equation 6.2.

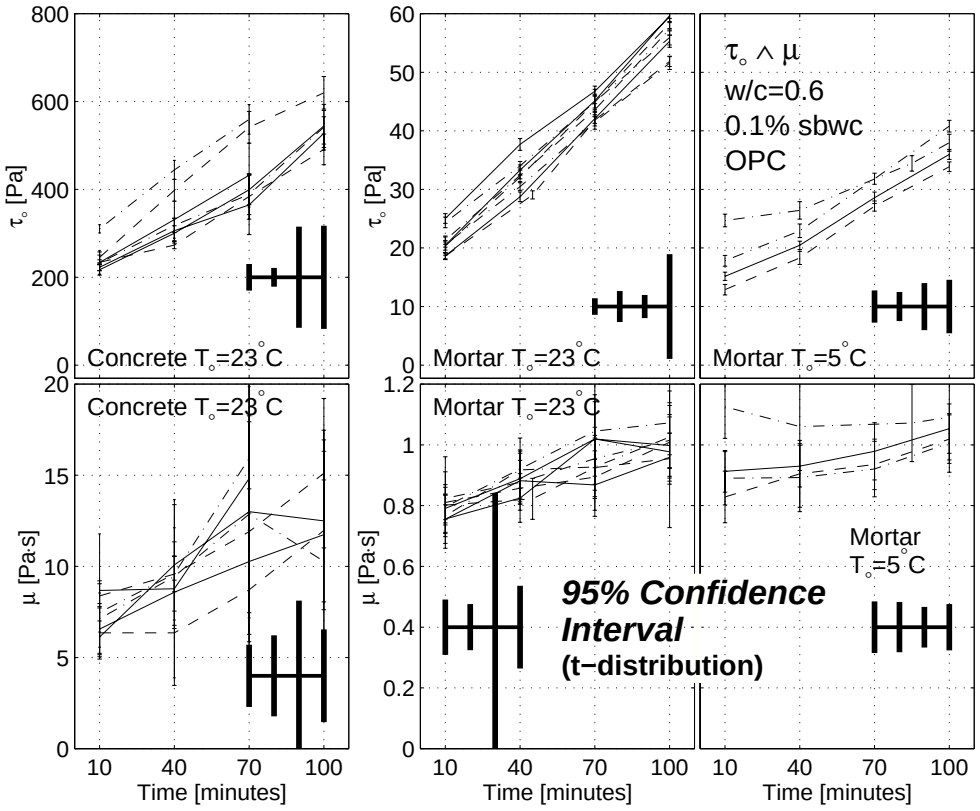


Figure 6.14: Measured yield value τ_o and plastic viscosity μ of concrete and mortar, using the different types of polymers ($w/c = 0.6$). The markings are shown in Figure 6.13.

Slump Value *S*

Figure 6.15 demonstrates measured slump value S for all batches of $w/c = 0.6$ [0.1% sbwc; OPC]. As mentioned before, due to the nature of the measurement, the calculation of confidence interval for a single slump measurement cannot be made. However, a 95% confidence interval calculated from the reproducibility tests in Chapter 5 can be made. The result is presented with four thick vertical lines, namely $\pm\Delta S = \pm t_{0.025} P_{\text{std}}(S)/\sqrt{n}$ (see Equation 6.2), connected with a horizontal line. As always, the lines are ordered in a chronological order from left to right and apply

at 10, 40, 70 and 100 minutes. These lines are generated from the mix using the **HMW Na** polymer (see the left illustration of Figure 5.4).

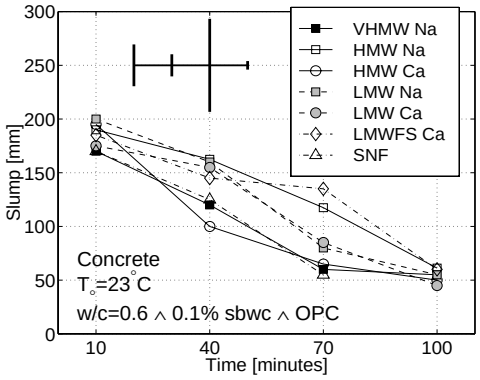


Figure 6.15: Measured slump value of concrete, using the different types of polymer.

From Figure 6.15 the same conclusion can be made as for the case in Figure 6.13, namely that there is not much significant difference between the different batches. This is likely due to the low amount of plasticizing polymer used (0.3% → 0.1% sbwc) in combination with high w/c-ratio ($w/c = 0.5 \rightarrow 0.6$). Ignoring the confidence interval $\pm\Delta\breve{\tau}_o$, the **HMW Na** and **LMWFS Ca** seems to give the best results, i.e. the lowest workability loss. Similar results are also produced with the ConTec viscometer. However, there seems to be a larger spread between the results in slump relative to the results of yield value τ_o. Nevertheless, there is an overall good correspondence between the results in Figure 6.13 and in Figure 6.15. This is shown in the left illustration of Figure 6.16. The right illustration demonstrates the somewhat weaker relationship between the slump value and the plastic viscosity μ.

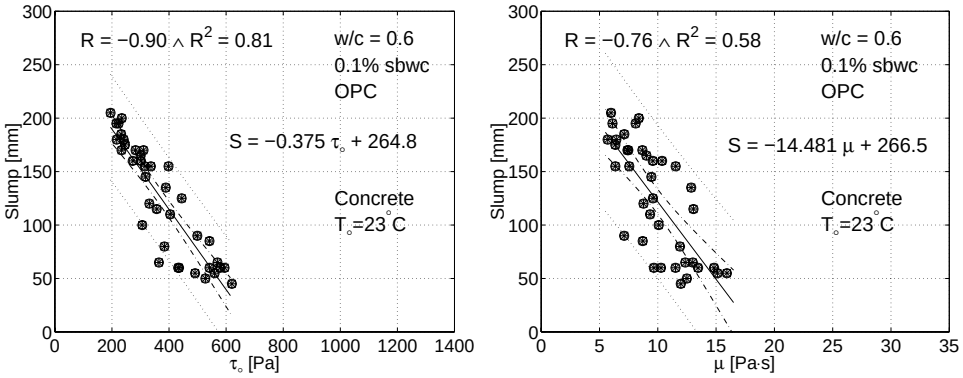


Figure 6.16: Comparison of measured slump value versus the measured yield value τ_o (to the left) and plastic viscosity μ (to the right).

As discussed in Section 6.2.1, the dashed dotted line in Figure 6.16, represents the 95% confidence interval for the regressed lines $S = b_1\,\tau_o + a_1$ and $S = b_2\,\mu + a_2$ and the dotted lines represents the 95% prediction interval.

6.2.5 0.6% sbwc ∧ 0.3% sbwc; w/c = 0.57; FAC

As mentioned in Section 5.4, to maintain reproducibility, it was necessary to modify the mix design when using the fly ash cement (FAC); i.e. the mix design of FAC-mortar is not calculated from a mix design of a concrete.

When going from Section 6.2.1 to Sections 6.2.3 and 6.2.4, several parameters are changed simultaneously, namely the w/c-ratio, the amount of plasticizing polymer and the amount of binders (see also Table 4.4, Page 78). Such approach is generally applied in ready-mixed plants. This makes it a bit harder to compare the results between the different sections. For example, comparing the effect of 0.6% sbwc dosage of polymer (Section 6.2.1) with dosage of 0.3% sbwc (Section 6.2.3) becomes a bit hard, because both the w/c-ratio and the amount of binders are also changing at the same time. With this in mind, both the w/c-ratio and the binder content is now kept constant for all the FAC-cases. Only the amount of polymer is now changed (see Table 4.6).

Yield Value τ_o

Figure 6.17 shows the measured yield value τ_o for all batches at $w/c = 0.57$ (FAC). In the two top illustrations at $T_o = 23°C$ (to the left) and $T_o = 38°C$ (to the right), the dosage of plasticizing polymer is 0.6% sbwc. However, for the two bottom illustrations, the dosage is reduced down to 0.3% sbwc.

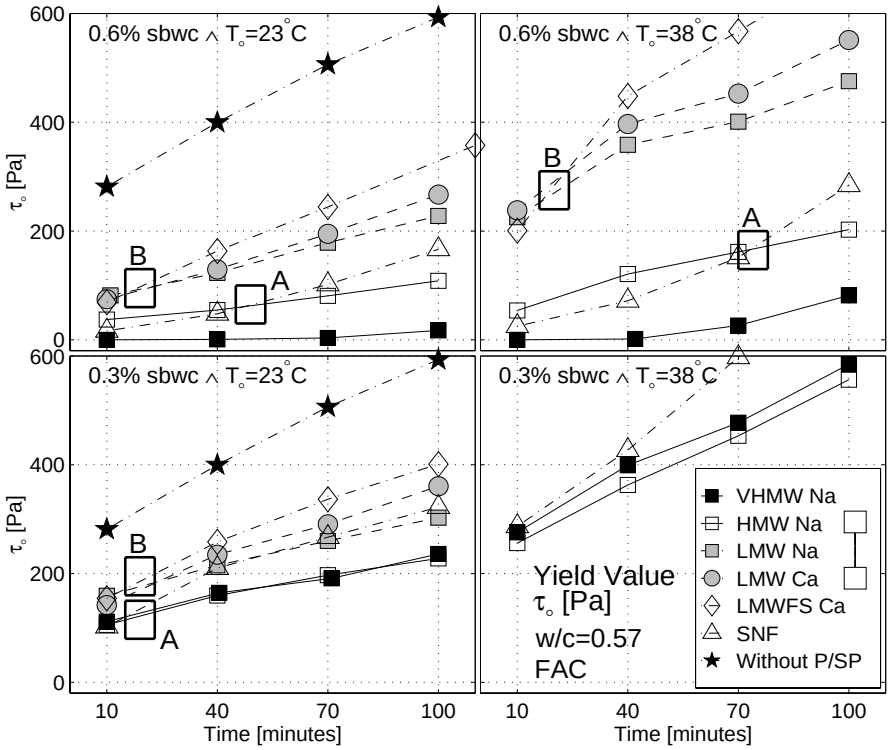


Figure 6.17: Measured yield value τ_o of mortar, using the different types of polymers ($w/c = 0.57$). The same results are shown in Figure 6.18 with error-bars and without markings. The corresponding result of plastic viscosity μ is shown in Figure 6.19.

Figure 6.18 shows the same results as Figure 6.17, however with the error-bars

$\pm\Delta\tau_o$ (Equation 6.1) and the four vertical lines $\Delta\breve{\tau}_o$ (Equation 6.2), and without markings. As applies in Section 6.2.1, the error-bar $\pm\Delta\tau_o$ of each measurement in Figure 6.18 represents 95% confidence interval, calculated according to Equation 6.1. The four thick vertical lines $\pm\Delta\breve{\tau}_o$ also shown, represents the 95% confidence interval calculated from the four repeated batches using the **HMW Na** polymer (extracted from the reproducibility tests of Section 5.4). The $\Delta\breve{\tau}_o$ value is calculated according to Equation 6.2. For example, in the case of mortar at 0.6% sbwc and $T_o = 23°C$ (the top left illustration in Figure 6.18), these lines are calculated from the standard deviation shown in the top left illustration of Figure 5.13 (Page 97), with $\nu = n - 1 = 4 - 1 = 3$ degrees of freedom. As shown in Figure 6.18, the difference in yield values between the batches of different polymers, are clearly significant.

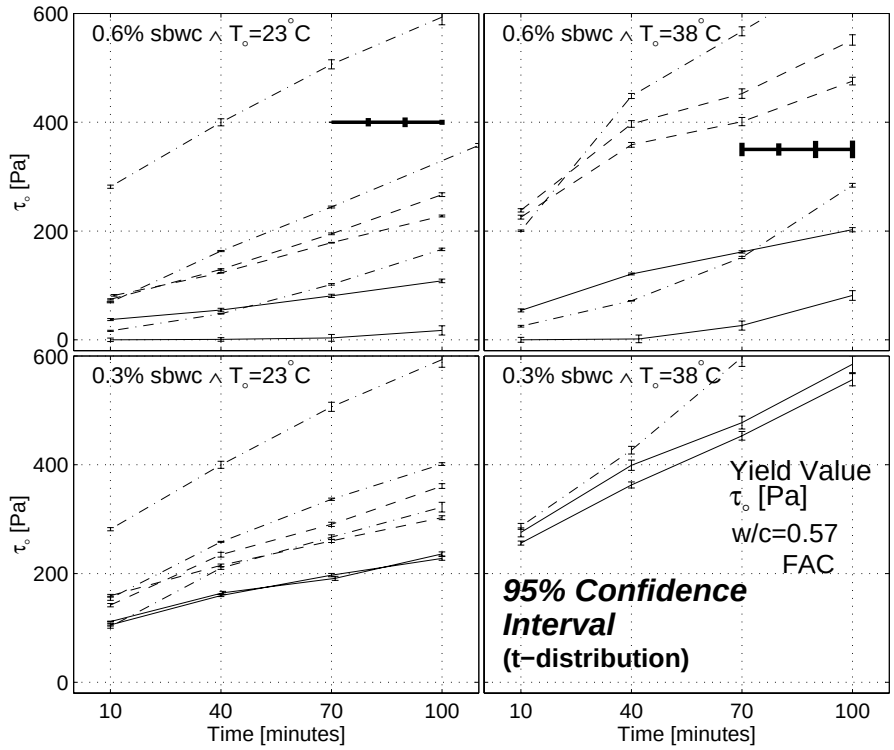


Figure 6.18: Measured yield value τ_o of mortar, using the different types of polymers ($w/c = 0.57$). The same results are shown in Figure 6.17 with markings and without error-bars. The corresponding result of plastic viscosity μ is shown in Figure 6.20.

Comparing the top left illustration shown in Figure 6.17 [0.6% sbwc; $T_o = 23°C$; $w/c = 0.57$; FAC], with center bottom illustration in Figure 6.1 [0.6% sbwc; $T_o = 23°C$; $w/c = 0.4$; OPC], the same rearrangement between different batches is produced. This is shown for example with the crossing of the different batches, marked with the capital letters **A** and **B** in both illustrations. The same type of observations exists also for the top right illustration shown in Figure 6.17 ($T_o = 38°C$), except for that the **SNF**-batch is less temperature sensitive, than what is shown in Figure 6.1.

The tendency of that the **VHMW Na** polymer performs equally as well as the **HMW Na** polymer, when the polymer dosage is reduced from 0.6% sbwc down to 0.3% sbwc, is produced in Figure 6.17. As mentioned previously, this trend is also

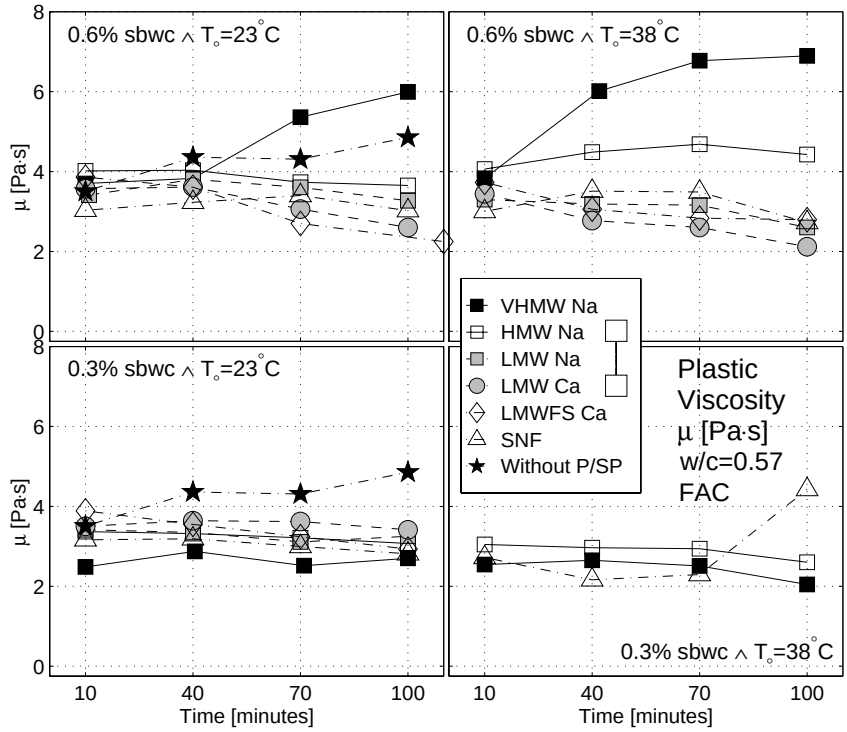


Figure 6.19: Measured plastic viscosity μ of mortar, using the different types of polymers ($w/c = 0.57$). The same results are shown in Figure 6.20 with error-bars and without markings. The corresponding result of yield value τ_o is shown in Figure 6.17.

produced when using the OPC. This is shown when comparing the case in Figure 6.1 [0.6% sbwc; $w/c = 0.4$; OPC] with Figure 6.7 [0.3% sbwc; $w/c = 0.5$; OPC].

Of the lignosulfonates, the general trend for the two top illustrations in Figure 6.17, is that the there are three distinct different groups. First is the batch using the **VHMW Na** polymer. Second group consist of the **HMW Na**-batch. The third group consist of the batches using **LMW Na**, **LMW Ca** and **LMWFS Ca** polymers, giving the set of largest yield values measured. This type of grouping is also produced in some of the illustrations in Figure 6.1. For the bottom left illustration in Figure 6.17, the **VHMW Na**-batch is merged into the same group as the **HMW Na**-batch, hence only two groups exists for that particular case. This type of grouping is also produced in some of the illustrations in Figure 6.7. Because of a very large yield value τ_o that applies for the mix of 0.3% sbwc at $T_o = 38°C$, only few batches where tested for that case, namely the **SNF**, **VHMW Na** and **HMW Na**-batches.

Plastic Viscosity μ

Figure 6.19 shows the measured plastic viscosity μ for all batches at $w/c = 0.57$ (FAC). As before, in the two top illustrations at $T_o = 23°C$ (to the left) and $T_o = 38°C$ (to the right), the dosage of plasticizing polymer is 0.6% sbwc, while it is 0.3% sbwc for the two bottom illustrations. Figure 6.20 shows the same results, however with the error-bars $\pm\Delta\mu$ (Equation 6.1) and the four vertical lines $\Delta\breve{\mu}$ (Equation 6.2), and without markings.

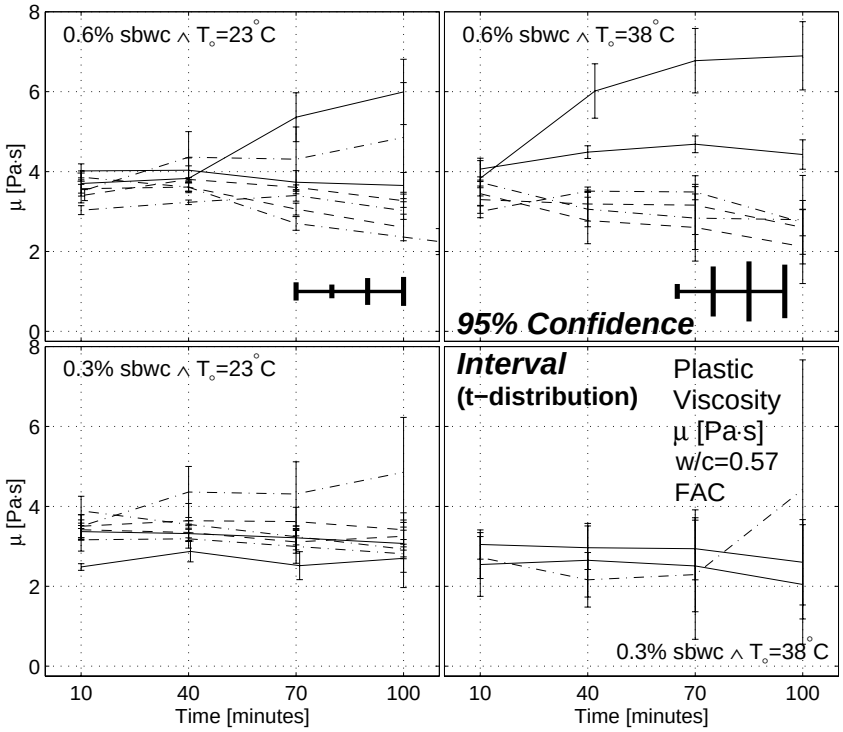


Figure 6.20: Measured plastic viscosity μ of mortar, using the different types of polymers ($w/c = 0.57$). The same results are shown in Figure 6.19 with markings and without error-bars. The corresponding result of yield value τ_o is shown in Figure 6.18.

As shown in Figure 6.20, there is not much difference between the measured plastic viscosity μ. As such, there is not much basis for discussing the effects of the polymers on plastic viscosity μ in Figure 6.19. However, one interesting phenomenon is to be noted in this figure (especially for the $T_o = 38°$C-case) and has been observed and discussed previously (see Page 113): The lower the yield value τ_o is, the higher the plastic viscosity μ becomes.

6.3 Discussion of Results: Concrete

6.3.1 Slump and Yield Value τ_o

Abrams slump test is used extensively in site work all over the world. Because of its simplicity, it remains the most widely used test for characterizing concrete consistency. Hence, it is important to demonstrate a relationship between the measured slump S and the viscometric values given by the ConTec **BML** Viscometer 3. There have been several attempts to find such relationship for other viscometers. Tanigawa et al. [120, 122] have proposed a graph showing the relationship between the slump and the viscometric values τ_o and μ. There it is evident that the slump value is more sensitive to the yield value τ_o than to the plastic viscosity μ. This result is in an agreement with Figures 6.6, 6.12 and 6.16, which shows that there is a considerable stronger relationship between yield value τ_o and slump than between plastic viscosity μ and

slump.

Murata et al. [83] have proposed an equation that relates slump only with yield value τ_o. This relationship is shown in Equation 6.3 below and is based on numerous experiments, where the yield value is extracted from a coaxial cylinders viscometer.

$$\tau_o = 714 - 473 \log(S/10) \tag{6.3}$$

The unit of slump in the above equation is (as always) in mm. According to de Larrard [25] it can be shown from a dimensional analysis that the slump is governed by the quantity $\tau_o/(g\,\rho_{sg})$, where g is gravity (see Equation B.26, Page 387) and $\rho_{sg} = \rho/\rho_{ref}$ is the specific gravity. The term ρ is the density of the fresh concrete [kg/m^3] and ρ_{ref} is the density of a reference fluid [36], which is (in this case) the density of water at 4°C, equal to 1000 kg/m^3. In his doctoral thesis, Hu [50] proposed a relationship between slump and yield value τ_o that includes density ρ, shown with Equation 6.3. This equation is based on some number of numerical simulations.

$$S = 300 - 270\frac{\tau_o}{\rho} = 300 - 0.270\frac{\tau_o}{\rho_{sg}} \tag{6.4}$$

In a NIST report [31] Ferraris and de Larrard suggests a modification to Equation 6.4, based on some numerous experiments, using the BTRHEOM viscometer (see Figure 1.2, Page 4). This modification is given with the equation below.

$$S = 300 - 0.347\frac{(\tau_o - 212)}{\rho_{sg}} \tag{6.5}$$

In his textbook [25], de Larrard adopts this equation for the range of slump higher than 100 mm. With a slight modification, the above equation can be applied for the data shown in the left illustration of Figure 6.6 [0.6% sbwc; $w/c = 0.4$; OPC]. More specifically, the following relationship can be extracted from this illustration:

$$S = 300 - 0.416\frac{(\tau_o + 394)}{\rho_{sg}} \tag{6.6}$$

The above equation falls within the boundary of the two dashed dotted line shown in Figure 6.6 (this slump function is also plotted in Figure 6.21). The reason for the use of "$(\tau_o + 394)$" instead of "$(\tau_o - 212)$" is because of the difference in yield value τ_o that the BTRHEOM viscometer and the ConTec **BML** Viscometer 3 produces. As shown in Figure 1.3 (Page 4), for the given concrete batch, the former viscometer produces a larger yield value than the latter.

When considering the results for the $w/c = 0.5$ and the $w/c = 0.6$ case, Equation 6.6 becomes somewhat less valid. This is apparent when looking at the left illustration of Figure 6.21, where all slump results of Figures 6.6, 6.12, and 6.16 are combined in one illustration. There it is demonstrated how the three different mix design of $w/c = 0.4$, $w/c = 0.5$ and $w/c = 0.6$ gives a slightly different relationship between slump and yield value τ_o. The reason for this could be related to the different amount of binders used in the three different mix design. The amount binders used in each mix design is shown in Table 4.3 and in Figure 6.21. With lower binder content and hence a closer distance between the larger aggregates, a self-bearing network of aggregates could form more easily during the "slump-flow". This self-bearing network inhibits further slump-flow that gravity g cannot over win. This network is similar to the bridging effect shown in Figure 3.8 (Page 62), except that the engine of the

viscometer can brake up such formation, when gravity cannot in a slump-flow. With this in mind, a small addition $-0.0077\,(\tau_o - 200)(320 - V_b)$ is supplied to Equation 6.6. The resulting equation is shown below.

$$S = 300 - 0.416\frac{(\tau_o + 394)}{\rho_{sg}} - 0.0077\,(\tau_o - 200)(320 - V_b) \tag{6.7}$$

The term V_b in Equation 6.7, is the amount of binder used in the concrete mix and its unit is in l/m^3. For the $w/c = 0.4$ batch, the amount of binder is $V_b = 320\,l/m^3$ (Table 4.3, Page 78), hence $-0.0077\,(\tau_o - 200)(320 - V_b) = 0$, and no addition is applied into Equation 6.6 for that case (see Footnote 2, Page 77 about binder).

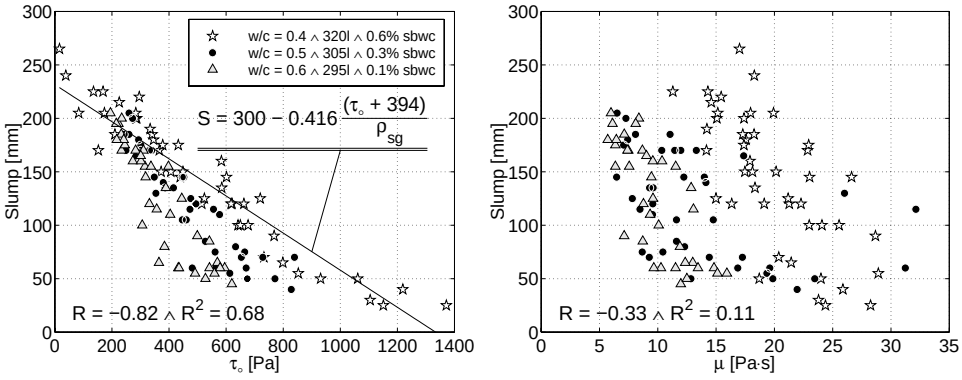


Figure 6.21: Comparison of measured slump value versus the measured yield value τ_o (to the left) and versus the plastic viscosity μ (to the right). All values are extracted from Figures 6.6, 6.12, and 6.16. The correlation coefficient R is of all data points.

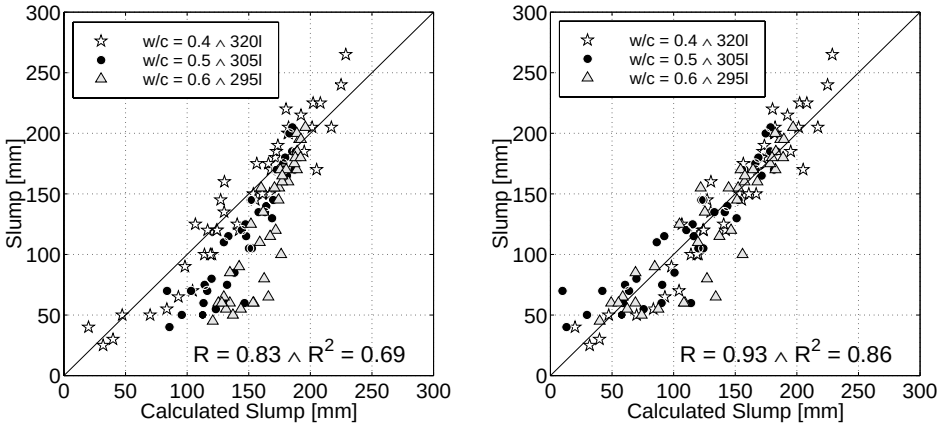


Figure 6.22: Comparison between measured slump and calculated slump by Equation 6.6 (to the left) and Equation 6.7 (to the right).

Plotting the measured slump versus the calculated slump by Equation 6.6, results in the left illustration of Figure 6.22. The type of deviation from the one-to-one correspondence shown in this illustration, is also reported by de Larrard [25] using the BTRHEOM viscometer. In that case, the deviation is related to the amount of plasticizing polymer used, rather than to the amount of binder. More specifically,

batches without plasticizers gave a deviation of calculated slump, while batches with plasticizers did not.

The right illustration of Figure 6.22 demonstrates the effect of the correction $-0.0077\,(\tau_o - 200)(320 - V_b)$ that is applied with Equation 6.7. With $R^2 = 0.86$, this illustration demonstrates that it is possible to gain a well established relationship between the Abrams slump and the yield value τ_o measured by the ConTec **BML** Viscometer 3. This good relationship exists in spite of the problem of gravel migration (see Chapter 10) that occurs for the concrete batches of this thesis.

6.3.2 Relationship Between Mortar and Concrete

As mentioned in Section 4.3.2, the mix design of the $0 - 2\,\text{mm}$ OPC-mortars used in this thesis, are calculated directly from the mix design of the corresponding concrete mixes. As such, it is to be expected that some kind of relationship exist between their measured viscometric values. Figures 6.23 and 6.24 demonstrates the degree of such relationship. Here, the terms τ_o^c and μ^c designates the viscometric values of the concrete. Likewise, the terms τ_o^m and μ^m designates the viscometric values of the corresponding mortar.

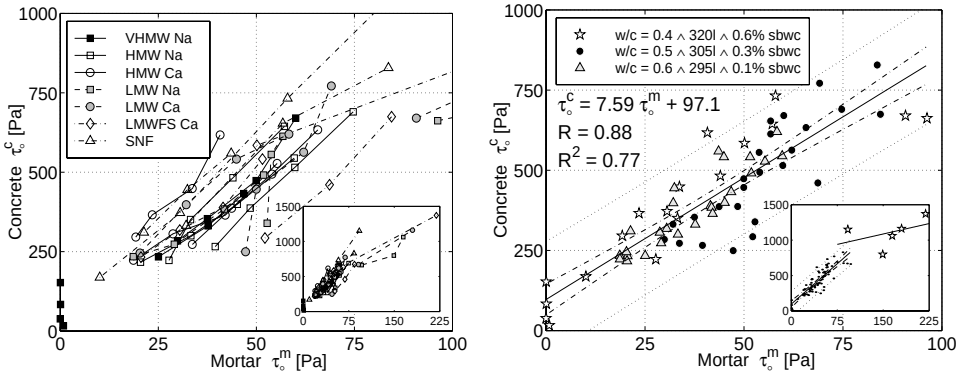


Figure 6.23: Relationship between measured yield value of concrete τ_o^c and mortar τ_o^m. Both illustrations demonstrate the same results, however plotted differently. The left plot is relative to the type of polymer used and the right plot relative to the mix design.

Figure 6.23 demonstrates the relationship between measured yield value of concrete τ_o^c and mortar τ_o^m. The dashed dotted line shown in the right illustration, represents the 95% confidence interval for the regressed lines $\tau_o^c = 7.59\,\tau_o^m + 97.1$ and the dotted lines represents the 95% prediction interval (see Section 6.2.1). As shown in this figure, there exits a good relationship between the measured yield value of concrete τ_o^c and of mortar τ_o^m. However, when the yield value of mortar τ_o^m exceeds about 100 Pa, an underestimated yield value of the concrete τ_o^c exists, relative to the straight line of $\tau_o^c = 7.59\,\tau_o^m + 97.1$. Combined error generated by plug flow (Section 3.5.3) and air entrainment (Section 5.5.2) for mortar, and the error generated by the gravel migration for concrete (Chapter 10), could be responsible for this. When calculating the correlation coefficient $R = 0.88$, shown in Figure 6.23, the five yield points shown in the small incorporated illustration are not taken into account.

Figure 6.24, demonstrates the relationship between measured plastic viscosity of concrete μ^c and mortar μ^m. Unfortunately, there is not a clear relationship between

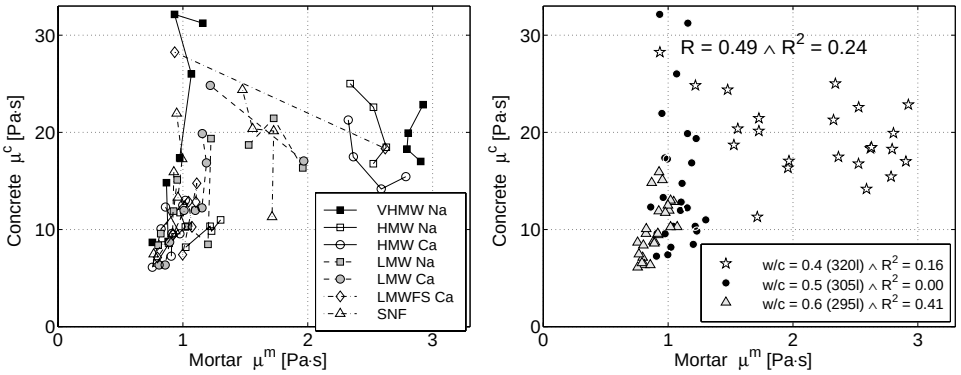


Figure 6.24: Relationship between measured plastic viscosity of concrete μ^{c} and mortar μ^{m}. Both illustrations demonstrate the same results, however plotted differently. The left plot is relative to the type of polymer used and the right plot relative to the mix design.

μ^{c} and μ^{m} as exists for $\tau_{\mathrm{o}}^{\mathrm{c}}$ and $\tau_{\mathrm{o}}^{\mathrm{m}}$. For the $w/c = 0.4$ case, the results are quite spread, and for the $w/c = 0.5$, a value of $R^2 = 0$ is a reality.

6.4 Discussion of Results: Mortar

It should be noted that terminologies from Sections 2.4 and 2.5 are very much used in the rest of this chapter. The subjects of these two sections are summarized in Section 2.6.3. Also, in accordance to the definition done in Footnote 25 (Page 38), when referring to a "cement particle", it is meant both its hydrated and its unhydrated part (see Figure 2.17, Page 38).

Quantity of Data Available

Because of the much larger quantity of data available for the mortar case, those will be used when considering the effects of the different (super)plasticizers on viscometric values, relative to the different conditions. In some cases, when appropriate, the concrete results is also included for comparison.

In Section 6.3.2, it was established that there is a certain relationship between the yield value τ_{o} of concrete and of mortar (see Figure 6.23). Hence, the results of yield value presented in the following sections can be extrapolated to the concrete, which again can be extrapolated to the slump value, as shown with Figure 6.22. Unfortunately, as shown with Figure 6.24, the relationship between the plastic viscosity μ of concrete and of mortar is not so clear. This poor correlation is most likely somewhat related to the error generated by plug flow (see Section 3.5.3) and air entrainment (see Section 5.5.3) from the mortar case. As has been previously mentioned, the plastic viscosity μ is much more sensitive to these errors, relative to the yield value τ_{o}. Hence a larger scatter in data results in Figure 6.24, relative to Figures 6.23. Gravel migration in concrete (Chapter 10) is probably also somewhat responsible for the overall scatter of the data shown in Figures 6.23 and 6.24.

When considering the temperature effects, then in each illustration it will always be referred to the ambient temperature condition T_{o}. But as discussed in the end of each Sections 5.3.3, 5.3.4 and 5.3.5, the actual temperature T of the mortar batches

deviates, at the most, only by few degrees from the ambient temperature T_o. Therefore, the ambient temperature will be sufficient descriptor of the actual temperature of the mortar batches (this applies for the whole Chapter 6). For further readings about the temperature, see the end of each Sections 5.3.3, 5.3.4 and 5.3.5.

Error Generated by Plug Flow (in Mortar)

As mentioned in Section 3.5, under sufficiently large ratio of yield value versus plastic viscosity τ_o/μ a plug will start to occur in the mortar sample, at the outer cylinder of the viscometer. As the rotational frequency f_o is reducing, the plug state will propagate towards the inner cylinder as shown with the right illustration of Figure 3.10 (Page 66). Depending on the τ_o/μ-ratio, this flow phenomenon introduces an error to the viscometric values τ_o and μ of mortar, when retrieved by Equations 3.26 and 3.27. However, as discussed in Section 3.5.3, for $\tau_o/\mu \propto 100\,\mathrm{s}^{-1}$, the error produced is only about order of magnitude of 1% for the yield value τ_o and of 10% for the plastic viscosity μ. As will be apparent in the following sections, the condition $\tau_o/\mu \leq 100\,\mathrm{s}^{-1}$ will apply in most cases and hence error generated by plug is generally not a real problem. However, in a few occasions, this ratio becomes as high as around $300\,\mathrm{s}^{-1}$. For example, this applies for some of the FAC-cases at 38°C. Under such condition, a larger error results, namely order of magnitude of 10% and 100% for the yield value τ_o and plastic viscosity μ, respectively. Fortunately, such cases are rare.

Error Generated by Air Entrainment (in Mortar)

As mentioned in Sections 5.5.2 and 5.5.3, air entrainment has a minor effect on the yield value τ_o of mortar. Unfortunately, the same does not apply for the measured plastic viscosity μ. Rather, from Figures 5.14 [$w/c = 0.4$; OPC] and 5.16 [$w/c = 0.57$; FAC] it appears that with increasing air content, the plastic viscosity μ decreases (as applies for the concrete case c.f. [144, 146]). The resulting percentage error present in plastic viscosity μ for each mortar batch is unknown, meaning that care must be taken when comparing plastic viscosity of two or more mortar batches. However, as is apparent in Figure 5.15 (Page 99), there exists a strong air stability for the case of [$w/c = 0.5$; OPC] in mortar, at least in the yield value range of about $\tau_o \in [0, 160]$ Pa, and for a moderate increase in yield value (as a function of time). Therefore, for this case, the plastic viscosity evolution μ can be considered reliable, when considering the effect of different polymer type used. There are two exceptions for this, namely the **SNF**-case and the Without P/SP-case, at the temperature of 38°C. These two cases are discussed on Page 113.

Given the small influence of the plug flow and air entrainment on the yield value τ_o for mortar, the error-bars $\pm\Delta\tau_o$ and $\pm\Delta\breve{\tau}_o$ (from Section 6.2.2) can be considered to include these errors. However, given the large influence of plug flow and air entrainment on the plastic viscosity μ, the error-bars $\pm\Delta\mu$ and $\pm\Delta\breve{\mu}$ might be somewhat underestimated.

Yield Value τ_o versus Plastic Viscosity μ

When comparing the results of Figure 6.2 with Figure 6.4, Figure 6.8 with Figure 6.10, and Figure 6.17 with Figure 6.19, three general results are observed:

1. **Polymer Type:** The effect of the different polymer types are mainly reflected in the measured yield value τ_o, rather than in the measured plastic viscosity μ.

Such result has been reported elsewhere when using lignosulfonates [128].

2. **Polymer Dosage:** Increased lignosulfonate addition reduces the yield value τ_o, while the plastic viscosity μ is much less affected. This result is in an agreement with other findings when using lignosulfonates [128, 146].

3. **Time Effect:** The plastic viscosity μ remains more or less a constant, or is even reducing in some cases, as a function of time, while the yield value τ_o is steadily increasing. Using the same type of lignosulfonates as in this thesis, such tendency has been reported elsewhere [95].

From the above points, it is clear that the change in shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ is mostly reflected by the change in yield value τ_o, rather than by the change in plastic viscosity μ.

6.4.1 Increased Time ⇒ Increased Shear Viscosity η

Phase Volume Φ, Surface Roughness and Coagulation State $J_t^{tot} = J_t^p + J_t$

When creating the (limited) rheological model for the cement paste, presented in Section 2.4.1 (see Equation 2.25, Page 26), it became clear that (at least) the phase volume Φ, surface roughness and coagulation state $J_t^{tot} = J_t^p + J_t$, are the controlling material parameters in changing the value of shear viscosity $\eta = [\mu + \tilde{\mu}] + [\tau_o + \tilde{\tau}_o]/\dot{\gamma}$ as a function of time. As the cement paste is the far most chemically changing component in mortar[2], the same consideration can go for the shear viscosity η of mortar.

For the cement paste, inside the mortar, a viscometric contribution of the type $\tilde{\mu}$ and $\tilde{\tau}_o$ are more or less filtered out by the intensive (re)mixing[3] procedure of Box 5 and 6 (Page 92) just prior to a measurement. That is, the reversible coagulated cement particles, are more or less dispersed (i.e. $J_t \approx 0$) just before a viscometric measurement $\Rightarrow \tilde{\mu} \approx 0 \wedge \tilde{\tau}_o \approx 0$. Hence, the change in shear viscosity of cement paste (inside the mortar) is rather a result of a change in phase volume Φ, surface roughness and **permanent** coagulation state $J_t^{tot} \approx J_t^p + 0$, giving a shear viscosity of the type $\eta = \mu + \tau_o/\dot{\gamma}$ for the mortar (and concrete).

For each experiment that spans over only 20 seconds (see the left illustration of Figure 3.7, Page 61), it is not to be expected that the change in number of permanent junctions J_t^p is large. It is rather during the time period of about 30 minutes, between each experiment, that the number of permanent junctions J_t^p are gradually increasing as described with Equation 2.65 (Page 50). Of course, the number of reversible junctions J_t are also gradually increasing during this same time period, as described by the same equation. However, by the intensive (re)mixing procedure of Box 5 and 6 prior to a measurement, reversible coagulated cement particles, are more or less dispersed, giving the condition $J_t \approx 0$ just before and during a measurement; i.e. the value of J_t is always reset to zero before each experiment.

As the phase volume Φ, surface roughness and permanent coagulation state J_t^p of the cement paste, inside the mortar, is always **increasing with time** (due to

[2]For mortar, coagulation of the smallest aggregate particles, say below 20 μm in radius, can also be an issue. This is because of how the van der Waals attractive potential energy V_A is present between any two particles in close vicinity of each other. However, because of the smaller quantity of the $0 - 100\mu$m aggregates: $0.1 \cdot 1200\,kg/m^3 = 120\,kg/m^3$ relative to the $600\,kg/m^3$ of cement (see Figure 4.4 and Table 4.4) the former will inflict the shear viscosity η to a less degree than the latter.

[3]The larger solid particles of the $0 - 2$ mm aggregates, could also give an additional dispersing and grinding effects on the (reversible) coagulated cement particles.

hydration), then so is also the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ of mortar. [A summary of how hydration changes the phase volume Φ, surface roughness and coagulation state $J_t^{tot} = J_t^p + J_t$, is given in Section 2.6.3].

When looking at a shear viscosity function like of $\eta = \mu + \tau_o/\dot{\gamma}$, the immediate question is: *Should both the internal parameters μ and τ_o also increase with time?* The answer to this is not straightforward. Assuming for a moment, that the plastic viscosity μ is only related to momentum exchange between free cement particles (i.e. not coagulated particles; see Figure 2.11, Page 27): As the phase volume Φ and surface roughness increases, a single momentum exchange between two free particles will also increase, which again will result in an increased contribution to the plastic viscosity μ. However, with increasing time, the number of free particles is decreasing as a result of their coagulation. Because of this decrease, the total momentum exchange between free particles could actually be reducing. If so, the plastic viscosity μ would decrease with increasing time. Assuming that the yield value τ_o is only related to momentum exchange between, and by, coagulated cement particles, the value of τ_o would always increase with time. This is because of how coagulation state J_t^{tot} is always increasing with time.

With always increasing phase volume Φ, surface roughness and coagulation state $J_t^{tot} \approx J_t^p$, the fundamental requirement is that the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ is increasing with time. How the internal parameter μ and τ_o should behave relative to each other, to make sure of this, is an another issue and not so straightforward.

Smaller Phase Volume Φ and Surface Roughness with Retardation

As mentioned in Section 4.2.2, every lignosulfonate batch contains also isolated sugar components (mainly pentose and hexose) that slows down hydration [105]. The slowing mechanism of cement hydration is generally known as retardation. The lignosulfonate molecule itself has also a retarding influence on hydration [105]. With these two slowing effects, the phase volume Φ and surface roughness will be smaller. From Section 2.6.3, this results in a lower shear viscosity η for mortar (and also for cement paste and concrete). However, too much retardation is undesirable. Apparently, the magnitude of the clinker mineral C_3A is very important in inhibiting a too long retardation, when using lignosulfonates (see Section 2.5.1 about the C_3A and C_3S). For example, the hydration of a pure C_3S-paste can be completely inhibited by the addition of calcium lignosulfonate [LS: 0.5 wt% of C_3S; $w/C_3S = 0.6$] [76]. In the absence of lignosulfonate, the pure C_3S-paste are well hydrated with or without the presence of C_3A [151]. In the presence of lignosulfonate and 2% of C_3A, it is reported for one particular case [151] that even after half a year, no hydration occurs [LS: 0.8 wt% of C_3S; $w/C_3S = 0.6$]. However, in this case, with the addition of 5% C_3A, hydration occurs within 14 days [LS: 0.8 wt% of C_3S; $w/C_3S = 0.6$] [151]. In the same reference, it is shown that the amount of adsorbed lignosulfonate on the C_3S-C_3A-particle, increases with increasing C_3A content (apparently through preferential adsorption onto C_3A) [151]. When most lignosulfonate had been adsorbed, the hydration rate of the paste was increased [151].

Smaller Coagulation State $J_t^{tot} \approx J_t^p$ with R_c, $1/\kappa + D_{pol}$ and V_T^{max}

As is shown with Equations 2.58 and 2.59 (Page 45), there are (at least) three variables that controls the coagulation rate H. These are the "radius" of the cement particle R_c, the maximum value of total potential energy V_T^{max} and its reach $2/\kappa + 2D_{pol}$

(see Page 44, about the reach). Increased coagulation rate H, results in an increased coagulation state J_t^{tot} (c.f. Equation 2.65, Page 50), which again results in an increase shear viscosity η. The relationship between J_t^{tot} and η is summarized in Section 2.6.3.

Zeta Potential ζ and Ionic Strength I

Increased zeta potential ζ (an approximation of the Stern potential ψ_δ) gives a larger electrostatic potential energy V_R (Equation 2.52, Page 40) and hence a larger energy barrier V_T^{max} (c.f. $V_T = V_R + V_A + V_S$). This results in a decreased coagulation rate H (Equation 2.59, Page 46) and hence in a smaller coagulation state J_t^{tot} (Equation 2.65, Page 50). In accordance with Section 2.6.3, this gives a lower shear viscosity η.

As shown in the left illustration of Figure 2.19 (Page 42), increased ionic strength I gives a reduced energy barrier V_T^{max} and a smaller reach $2/\kappa$. This results in a larger coagulation rate H and hence in an increased coagulation state J_t^{tot}, which is then observed with a larger shear viscosity η.

In the limitation of the experimental work done here, neither ionic strength I nor zeta potential ζ measurements are made. However, experiments conducted at Borregaard LignoTech [103], give that no major changes in the ionic strength I is measured from 10 minutes to 1 hour after water addition (pore water extracted from concrete using OPC with lignosulfonates). Also, the zeta potential ζ during the first 90 minutes from water addition, is measured as a constant with time [39] (continuously stirred cement paste at $w/c = 0.5$ using OPC with lignosulfonates). The value of zeta potential ζ is only changed with different polymer type used in each batch [39]. With constant ionic strength I and constant zeta potential ζ as a function of time in each batch, there is seemingly no indication of a loss in electrostatic repulsive potential energy V_R that could parallel the measured increase in shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ [39]. This means, if the electric double-layer do provide an addition to the energy barrier V_T^{max} of relevant magnitude, it is a constant contribution with time (however, apparently different constant contribution from batch to batch, using the different type of polymer, c.f. the top right illustration of Figure 2.18, Page 39).

Steric Hindrance (V_S and D_{pol})

With adsorption of lignosulfonate polymers on the cement particle surface, the cement particles are inhibited to physically approach each other so closely as before. This effect is known as steric hindrance and is mathematically represented with the steric repulsive potential energy V_S. The maximum value of V_S could be related to the degree of polymer adsorption and to the compressive strength of the polymer. The reach of this potential energy, namely $2D_{pol}$, is dependent on the thickness of the adsorbed layer of polymers on the cement particle surface. For a given mono-layer adsorption, the value of D_{pol} represents the dimension of the polymer in the manner as shown with the right illustration of Figure 2.22 (Page 45). Hence, with increasing molecular weight values M_n and M_m (see Table 4.2, Page 74), the reach $2D_{pol}$ will increase. Illustration **C** in Figure 2.21 demonstrates how the steric repulsion V_S increases both the value of the energy barrier V_T^{max} and the reach $2/\kappa + 2D_{pol}$. This results in a lower coagulation rate H and hence in a lower shear viscosity η as discussed in Section 2.6.3.

Uchikawa et al. [136, 137] reports (in line with [151]) that lignosulfonates adsorbs more on the C_3A compound, relative to the C_3S compound. For this case, the thickness of the adsorbed layer is reported to be $D_{pol} = 100\,\text{nm}$ on the C_3S surface, while $D_{pol} = 200\,\text{nm}$ on the C_3A surface. With this, a multi-layer adsorption is considered

to apply, instead of a mono-layer adsorption [136]. A 100 layers has been reported in another particular case [105]. The degree of uneven adsorption between C_3S and C_3A, depends on the type of organic admixture used [135, 137, 138]. Elsewhere, it is not agreed that so many adsorbed layers exists in reality [85]: Adsorption is rather considered to consist of only few layers at the most. Additional layers above the few first ones, is rather defined as precipitation. The distinction between polymer adsorption and polymer precipitation, relays in the force keeping the polymers in place on the cement particle. Adsorption relies on attractive forces between the substrate (i.e. the cement particle solid surface) and the polymer, and will be substrate specific. For precipitation, it is rather polymer-polymer (attractive) forces that keeps the polymers in place. Precipitation is not substrate specific [85]. However, it is believed that both the adsorbed and the precipitated polymers always remains more or less on the cement particle [85] and hence in either case, do provide a same kind of contribution to the steric potential energy V_S. As such, the term "adsorption" will mean both a true adsorption and precipitation, in this thesis.

With increasing hydration, larger part of the polymers could be integrated into the solid surface (of the cement particle), as it is more and more covered by a hydrate (see Figure 2.22). This results in a decreased steric reach $2D_{pol}$ as a function of time, which again results in an increased coagulation rate H (see Equation 2.59, Page 46) that parallel the measured increase in shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ as a function of time. Other type of degradation of the adsorbing polymer could also be present, giving a larger loss in steric potential energy V_S and hence in lower energy barrier V_T^{max}. This would give a still larger coagulation rate H (Equation 2.59) and therefore an increased coagulation state J_t^{tot} (c.f. Equation 2.65) and hence larger shear viscosity η (c.f. Section 2.6.3).

Steric Hindrance V_S versus Electrostatic Repulsion V_R

The characteristic diameter of a lignosulfonate macromolecule is reported to be 10 nm [18]. Hence, the reach of the steric potential V_S is at least in the order of $2D_{pol} = 20$ nm assuming a mono-layer adsorption, and most likely larger since multi-layer adsorption seems to be a realty. With this in mind and looking at the left illustration of Figure 2.19, which demonstrates that the (effective) thickness of the diffuse layer $1/\kappa$ is always less than 1 nm, it is clear that the reach $2D_{pol}$ of the steric hindrance (V_S) is far greater than the reach $2/\kappa$ of the electrostatic repulsion (V_R). With this small reach (i.e. small $2/\kappa$), the relevance of the electrostatic potential energy V_R can be considered to be questionable. However, as will be apparent in the discussion around Figure 6.29, then for the smaller lignosulfonates **LMW Na** and **LMW Ca**, electrostatic contribution seems to be somewhat important. This is also discussed in Section 6.4.8.

Experimental Results

In the coming sections, the rheological results of the mortar experiments are presented. When discussing the results, the overall above theory will be used extensively. To begin with, the reference batches ("Without P/SP") are considered. Thereafter, the mortar batches using the different lignosulfonate types are presented.

6.4.2 Reference: Without P/SP

Before starting in examining the effects of the different types of lignosulfonates as a function of temperature and time, it is interesting to first consider the reference batches, namely the "Without P/SP" batches. These are the batches not using any plasticizers or superplasticizers.

Increased Temperature ⇒ Increased Shear Viscosity $\eta = \mu + \tau_o/\dot{\gamma}$

Generally, increased temperature accelerates the early rate of cement hydration, i.e. in the pre-dormant and dormant period [58]. This results in a still larger phase volume Φ and surface roughness. From Section 2.6.3, it is therefore to be expected that with increased temperature, the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ will also increase.

Without Polymers ⇒ Largest Shear Viscosity $\eta_{max} = \mu + \tau_o/\dot{\gamma}$

Without polymers adsorbed on the cement particle, opposite electrically charged surfaces of two cement particles becomes naked to each other, resulting in their electrostatic attraction; i.e. the electrostatic repulsion V_R, is either partially or completely replaced by an electrostatic attraction V_R^a. This results in a larger coagulation rate H than anticipated by the van der Walls attraction V_A alone. Also, without the polymers, no steric effects V_S are present to keep the cement particles physically apart. This results in an additional increase in coagulation rate H. From Equation 2.65 (Page 50), it is therefore to be anticipated that the coagulation state J_t^{tot} will be of the largest value, for mortars not using any polymers. In addition to this, without the retarding sugar that is present with the lignosulfonate polymer (mainly pentose and hexose, c.f. Section 4.2.2), cement hydration occurs more rapidly. This result in a still larger increase in phase volume Φ and surface roughness. With the particular relationship of surface roughness, phase volume Φ and coagulation state J_t^{tot} with the shear viscosity η (see Section 2.6.3), it is to be anticipated that the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ will be of the largest value for mortars not using any polymers.

Experimental Results

Figure 6.25 shows the viscometric results τ_o and μ, as a function of temperature and time when not using any polymers. Since the yield value τ_o is much larger than the plastic viscosity μ, the former is more accountable for the large magnitude in shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$. As shown in this figure, the yield value τ_o increases with increasing temperature and with increasing time. As mentioned above and in Section 6.4.1, this temperature and time dependency is anticipated. The plastic viscosity μ does however not show such a clear tendency. That is, the error-bars surmounts the small changes in the plastic viscosity μ as a function of time and temperature. However, in ignoring this, the case of $w/c = 0.5$ shows a somewhat reversed temperature dependency relative the yield value. That is, with increasing temperature, the plastic viscosity μ is (slightly) decreasing. The $w/c = 0.4$ results of plastic viscosity are more randomly oriented, with some kind of increase in plastic viscosity μ with increasing temperature. Given the influence of plug flow and air entrainment on the plastic viscosity, it is not wise to make any physical hypothesis about the plastic viscosity behavior, shown in Figure 6.25.

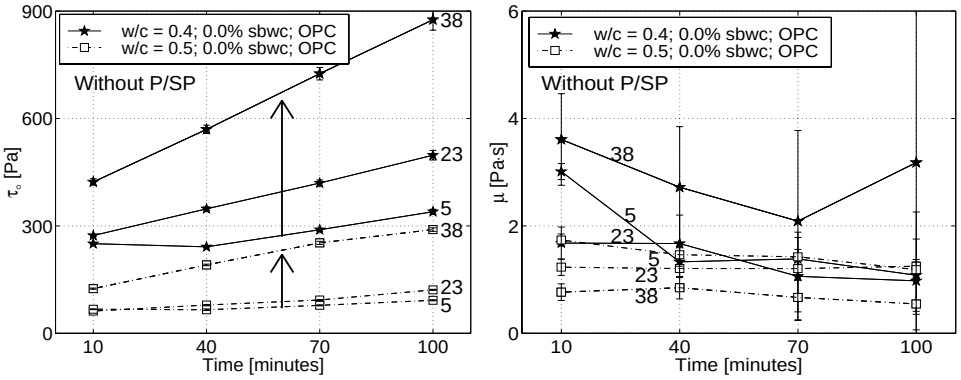


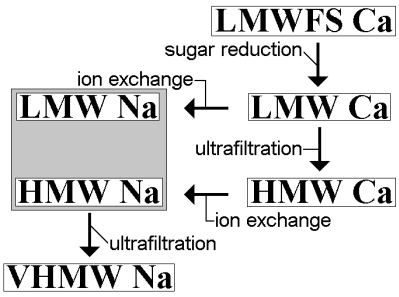
Figure 6.25: Measured yield value τ_o (to the left) and plastic viscosity μ (to the right) of mortar batches, not using any (super)plasticizers. The numbers 5, 23 and 38 demonstrates the corresponding temperature. The error-bars shown, are calculated according to Equation 6.1.

Summary of Main Results

1. The plastic viscosity μ is not significantly changing with time or temperature.
2. The yield value τ_o increases with increasing temperature and time. Since the plastic viscosity μ is more or less a constant, this yield value behavior is anticipated. This is because of the pre-mentioned postulate about that the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ must increase with time and temperature.

What now follows in the coming sections, is a comparison of viscometric values between the batches using the different polymer types. In most cases, the mix design is the same as for the batches shown in Figure 6.25. The exceptions are when using the FAC (mix designs of mortar are shown in Tables 4.4 (OPC) and 4.6 (FAC), Page 78). All the following figures are clearly labeled with OPC and/or FAC. Knowing how the "Without P/SP" batches behaves as a function of temperature and time, it is now interesting to examine how this behavior changes with the introduction of the lignosulfonates into the mortar batches.

6.4.3 HMW Na versus LMW Na



As mentioned in Section 4.2.2, the main difference between the **HMW Na** and the **LMW Na** polymer is that the former has gone under an ultrafiltration, while the latter has not (see Figure 4.2, Page 73). This means that their basic difference lies in the molecular weight distribution. In other words, the **HMW Na** polymer consist of larger molecules, relative to the **LMW Na** polymer. This is more specifically shown in Table 4.2 (Page 74), which demonstrates that there is a certain difference in their number average molecular weight M_{n} and their mass average molecular weight M_{m} (see Equation 4.1 for definition, Page 74). As is shown in Figure 4.3, this means that for the given dosage of 0.6% sbwc, a larger portion of the **HMW Na** polymer is adsorbed on the surface of the cement particles, relative to the **LMW Na** polymer. With the larger adsorption and larger dimension of the **HMW Na** polymer, a larger steric potential energy V_{S} results between the cement particles, relative to when using the **LMW Na** polymer. This gives a larger contribution to the energy barrier $V_{\mathrm{T}}^{\max}$ and reach $2D_{\mathrm{pol}}$ and therefore reduced coagulation rate H for the **HMW Na**-case (see Equation 2.59, Page 46). From Section 2.6.3, this results in a lower shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$.

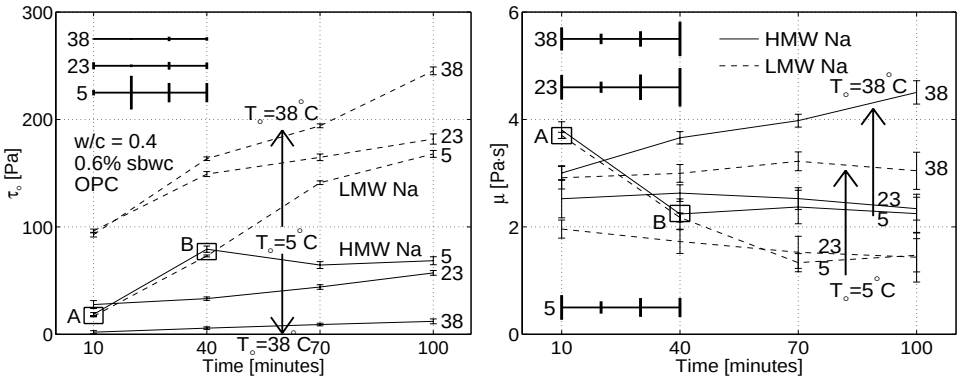


Figure 6.26: Measured yield value τ_{o} (to the left) and plastic viscosity μ (to the right) of mortar, using either the **HMW Na** or the **LMW Na** polymer (OPC). The numbers 5, 23 and 38 demonstrates the corresponding temperature.

The measured viscometric values of the mortars using either the **HMW Na** polymer or the **LMW Na** polymer are shown in Figure 6.26 (OPC). These results are extracted from Section 6.2.1. All the error-bars shown in this figure are the same ones as presented in Figures 6.2 and 6.4: The error-bars shown in the top left corner in both illustrations in Figure 6.26, represents 95% confidence interval generated from the reproducibility tests in Chapter 5 for mix using the **HMW Na** polymer. These error-bars are calculated according to Equation 6.2. However, the error-bars shown on each graph, are calculated according to Equation 6.1 (see Section 6.2.2).

The difference between the two above-mentioned polymer types is clearly reflected in the left illustration of Figure 6.26. There it is shown an opposite yield value τ_{o} behavior for the batches using the **HMW Na** polymer, relative to the ones using the

LMW Na polymer: With increasing temperature, the batches of the **HMW Na** polymer, gives an overall reduced yield value τ_o, while the batches of the **LMW Na** polymer give an increased yield value τ_o. The latter case has the same type of "natural" yield value evolution as is shown in Figure 6.25.

In most cases, the yield value τ_o is always increasing with time from water addition. The exception is the **HMW Na**-case at 5°C, in which between 40 and 100 minutes after water addition, no significant change is present (see also Figure 5.10, Page 95).

The temperature dependency of the **HMW Na**-batches, shown in Figure 6.26, are quite remarkable and is something not to be expected at first consideration. Even at the low dosage of 0.3% sbwc and at the higher water content of $w/c = 0.5$, there is still a small decrease in yield value with increasing temperature, when using the OPC. This is shown with the right illustration of Figure 6.27. However, this type of result is changed when changing the mix design from $[w/c = 0.4 \wedge 0.5;$ OPC] to $[w/c = 0.57;$ FAC]. This is shown in the same figure and demonstrates that when using the FAC at 0.6% or 0.3% sbwc, the "natural" yield value evolution starts to appear.

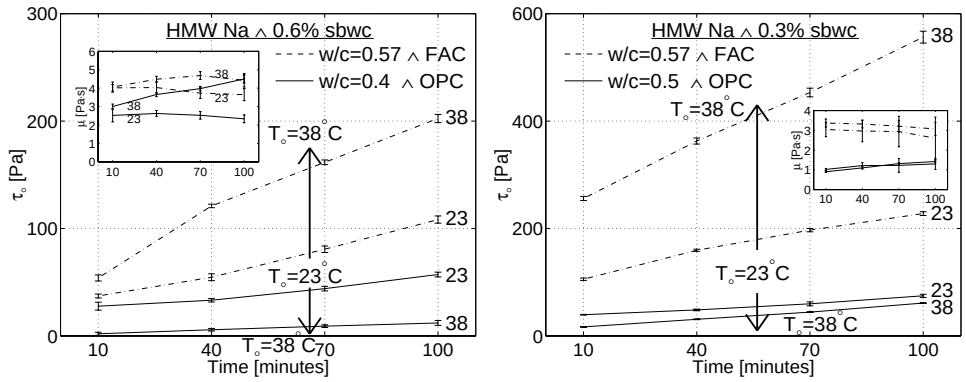


Figure 6.27: Measured yield value τ_o (the main illustrations) and plastic viscosity μ (small incorporated figures) for the FAC- and OPC-cases, using the **HMW Na** polymer. To the left is the dosage of 0.6% sbwc applied and to the right is the dosage 0.3% sbwc used.

Going back to Figure 6.26, an unexpected experimental result is made at 10 and 40 minutes at 5°C. This is marked with the capital letters **A** and **B** in both illustrations. The peculiar result consist of the fact that the same set of viscometric values τ_o and μ are produced for both cases of **HMW Na** and **LMW Na**. This is unexpected given the different properties of the two polymer types used in the different batches. This could be related to a possible slower adsorption process for the **HMW Na** polymer, relative to **LMW Na**, at the temperature of 5°C.

For most parts, the plastic viscosity μ in Figure 6.26 is either slightly increasing or simply is a constant as a function of time. The exception for this applies at 10 and 40 minutes, with temperature of 5°C. These two points are marked with the pre-mentioned capital letters **A** and **B**. In ignoring these two points, it is clear that the time dependency of the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ is mostly reflected by the yield value τ_o, rather than by the plastic viscosity μ. However, when considering the temperature dependency, then in either case of **HMW Na** or **LMW Na**, the plastic viscosity μ increases somewhat with increased temperature and hence also contributes to the temperature dependency of the shear viscosity η. Similar conclusion, as in the above, can be drawn for the FAC-batches in Figure 6.27.

The effect of plug flow and air entrainment on the plastic viscosity μ data, is considered in Sections 6.4.5 and 6.4.8.

Before making additional discussion for the overall above measured viscometric values, it is interesting to first consider the effects of the **HMW Ca** and the **LMW Ca** polymers at the same w/c-ratio and polymer dosage. This is the issue of next section. However, before this, a summary of the main results are given:

Summary of Main Results

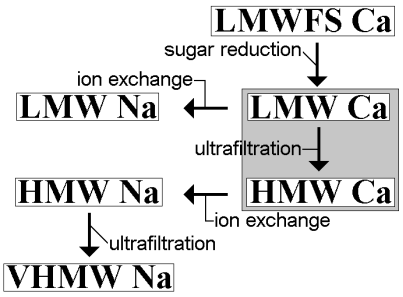
1. The **HMW Na** polymer is more effective plasticizer than the **LMW Na** polymer.

2. When using the **HMW Na** polymer at 0.6% sbwc [$w/c = 0.4$; OPC], a clear reduction in yield value τ_o exists for the corresponding batches, with increasing temperature. A smaller, and not so clear, reduction applies also when using the lower dosage of 0.3% sbwc [$w/c = 0.5$; OPC].

3. The above behavior is however not universal: When using the **HMW Na** polymer at 0.6% sbwc, with FAC ($w/c = 0.57$), a clear increase in yield value τ_o exists, with increasing temperature. The same result applies when using the dosage of 0.3% sbwc [$w/c = 0.57$; FAC].

4. The time dependency of the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ is generally more controlled by the yield value τ_o, rather than by the plastic viscosity μ. Some temperature dependency is present for the plastic viscosity μ (OPC, FAC).

6.4.4 HMW Ca versus LMW Ca



The main difference between the **HMW Ca** and the **LMW Ca** polymer is that the former has gone under an ultrafiltration, while the latter has not (see Figure 4.2). This means that their basic difference is the same as between the **HMW Na** and the **LMW Na** polymer in the previous section, and lies in their molecular weight numbers M_{n} and M_{m}. With the larger adsorption (Figure 4.3, Page 75) and the larger dimension (Table 4.2, Page 74) of the **HMW Ca** polymer, a larger steric potential energy V_{S} results between the cement particles, relative to when using the **LMW Ca** polymer. This gives a larger contribution to the energy barrier $V_{\mathrm{T}}^{\max}$ and reach $2D_{\mathrm{pol}}$ for the **HMW Ca**-case, and hence results in a smaller coagulation rate H (see Equation 2.59, Page 46). With a smaller coagulation rate H, a lower coagulation state $J_{\mathrm{t}}^{\mathrm{tot}} \approx J_{\mathrm{t}}^{\mathrm{p}}$ results (see Equation 2.65, Page 50). This is then observed with a lower shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$ (the relationship between $J_{\mathrm{t}}^{\mathrm{tot}}$ and η is discussed in Section 2.6.3).

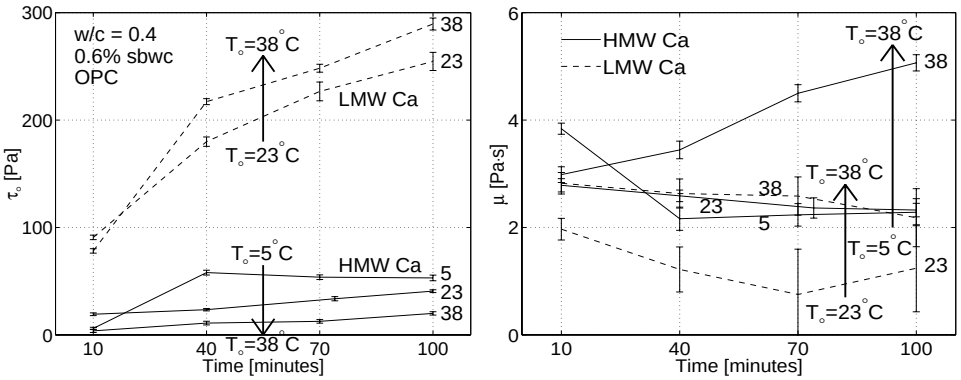


Figure 6.28: Measured yield value τ_{o} (to the left) and plastic viscosity μ (to the right) of mortar, using either the **HMW Ca** or the **LMW Ca** polymer (OPC). The numbers 5, 23 and 38 demonstrates the corresponding temperature.

The difference between the **HMW Ca** and the **LMW Ca** polymer is clearly reflected in Figure 6.28 (OPC). More precisely, the similarities between this figure and of Figure 6.26 are very large. Not only is there a same type of evolution in measured yield value τ_{o}, but the measured plastic viscosity μ is almost identical in both figures. This is visible with Figure 6.29, which demonstrates a comparison plot of the result shown in Figure 6.26 (**HMW/LMW-Na**) with the result shown in Figure 6.28 (**HMW/LMW-Ca**). If both last-mentioned figures where exactly identical, all measured points would fall on the straight line (with the slope 1) shown in both illustrations in Figure 6.29. Such condition applies almost for the measured plastic viscosity μ and to some lesser degree for the measured yield value τ_{o}. This demonstrates the overall large similarities between Figure 6.26 and Figure 6.28.

As is mentioned in Section 4.2.2, the only difference between the **HMW/LMW-Na** polymers and the **HMW/LMW-Ca** polymers is the use of so-called ion exchange in the manufacture at the Borregaard plant. In Figure 6.29, it is demonstrated that there is roughly an one-to-one correspondence between the **HMW/LMW-Na** results

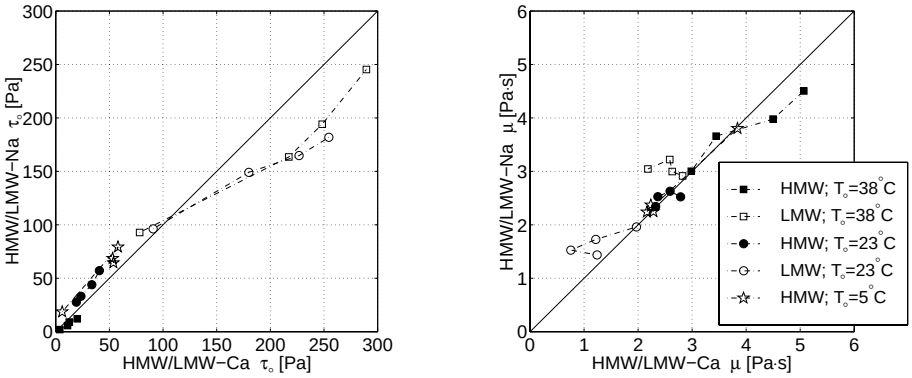


Figure 6.29: Comparison of the results presented in Figure 6.26 and in Figure 6.28. If both last-mentioned figure where identical, all measured points would fall on the straight line (with the slope 1) shown in both the left and the right illustration.

and the **HMW/LMW-Ca** results, especially for the plastic viscosity μ. However, when comparing the result between the **LMW Na**-case with the **LMW Ca**-case in a more detail, some larger yield value τ_o appears for the latter case. Such result has been reported elsewhere [105, 128]. With an (active) cation of valence $z_{Ca} = 2$ (**LMW Ca** $\rightarrow$ **LMW**$^{-2n}$ + n Ca^{+2}), a larger contribution to the ionic strength I is likely (see Equation 2.54, Page 40). As mentioned on Page 129, increased ionic strength I results in a larger coagulation rate H and hence in a larger shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ (reflected here by a larger yield value τ_o).

The ionic strength I can only influence the total potential energy V_T through the electrostatic potential energy V_R. When this last-mentioned potential energy is completely overshadowed by the steric potential energy V_S (resulting in $V_T = V_R + V_A + V_S \approx V_A + V_S$), ionic strength I will only influence coagulation rate H through $1/\kappa$. With a condition of $2/\kappa + 2D_{pol} \approx 2D_{pol}$ for the **HMW** products (c.f. Page 130), ionic strength I will in reality have no influence on the coagulation rate H. As such, ionic strength I will neither have any effect on the shear viscosity η (see Section 2.6.3 and 6.4.1). With the large molecular weight values M_n and M_m for the **HMW Na** and the **HMW Ca** polymers (see Table 4.2), it is likely that the above-mentioned *ionic-strength-independence* applies for the batches using these polymer types. In Figure 6.29, it is shown that the yield value τ_o for the **HMW Na**-case is generally higher than the **HMW Ca**-case. Relative to the overall above text, one should avoid in explaining this difference with an ionic strength I (or zeta potential ζ) considerations.

Now, it is clear from either Figure 6.26 or Figure 6.28 that at the dosage of 0.6% sbwc and $w/c = 0.4$, the high molecular weight products **HMW Na** and **HMW Ca** gives a reduced yield value τ_o with increasing temperature (OPC). Relative to the "natural" yield value evolution shown in Figure 6.25, this is an unexpected behavior. The reason for this unexpected yield value behavior, could lie in that the larger lignosulfonate molecules have some kind of swelling capabilities that increases with increasing temperature. If so, this would result in an enhanced steric potential energy V_S, at least by its increased reach $2D_{pol}$ from the solid surface of the cement particle. In a textbook by Nicholson [90], such swelling capabilites of crosslinked macromolecules is somewhat discussed, however not discussed relative to tempera-

ture changes. This swelling hypothesis is shown with the left part of Figure 6.30. Consisting of a great quantity of large crosslinked macromolecules, the **HMW Na** and **HMW Ca** polymers fits this profile (see Section 4.2.2 about the structure of the different lignosulfonate types).

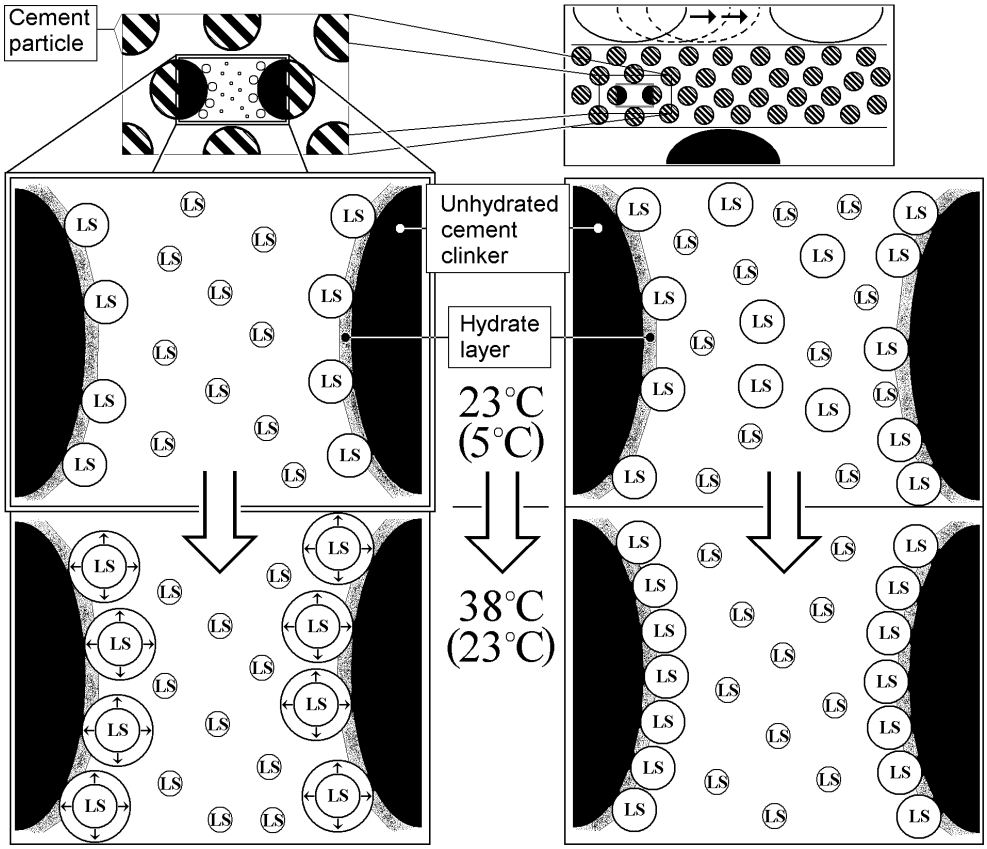


Figure 6.30: Schematic presentation of the swelling (to the left) and of the adsorption theory (to the right). The top right illustration is extracted from Figure 2.9 (Page 25). A more realistic presentation of the cement particle is shown in Figure 2.17 (Page 38). Adsorption is drawn here as a mono-layer. However as mentioned in Section 6.4.1, a multi-layer adsorption appears to be more likely. Regardless of multi-layer or mono-layer adsorption, the principle shown here applies.

The other hypothesis for explaining the yield value behavior of the **HMW Na**- and **HMW Ca**-batches in Figures 6.26 and 6.28, could be that the adsorbing capability of the polymers (on the cement particles), increases with the increasing temperature. In his doctoral thesis, Flatt [32] measures an increased adsorption of plasticizing polymers (however, not lignosulfonates) on MgO particles, with increasing temperature (pH = 12). In his work, he argue that with increased temperature, a larger number of oppositely charged ions (relative to the particle surface) desorb with increased temperature, leaving a larger surface area available for the polymers to adsorb on. This adsorption theory is shown in the right illustration of Figure 6.30. Such adsorption would enhance the steric repulsion V_S, giving an additional contribution to the energy barrier V_T^{max} against coagulation rate H. If this theory is valid, then it seem not to apply (or is at least less apparent) for the **LMW Na**- and **LMW Ca**-cases, as is

evident by the experimental results shown in Figures 6.26 and 6.28.

As mentioned before, it is clear from either Figure 6.26 or Figure 6.28 that at the dosage of 0.6% sbwc and $w/c = 0.4$, the high molecular weight products **HMW Na** and **HMW Ca** gives a reduced yield value τ_o with increasing temperature (OPC). However, as shown with Figure 6.27 for the **HMW Na**-case, this type of unexpected yield value behavior is not universal: When using the FAC at $w/c = 0.57$, the "natural" yield value behavior starts to occur; i.e. the yield value increases with increasing temperature. A possible start in understanding this, is by noticing the large quantity of water, relative to cement content, that is always present when applying the FAC, namely of $w/c = 0.57$. That is, the (relatively) large quantity of water could overshadow the pre-mentioned polymer behavior shown in Figure 6.30. Also, the fineness of the FAC is $437\,m^2/kg$ (Blaine), compared to the $340\,m^2/kg$ for the OPC (see Section 4.2.1). This means that a less mount of polymer is available per square meter of cement particle surface. Hence, the effect of the increased adsorption and/or swelling of the polymers could be overshadowed by a too large quantity of water and cement particle surface area, present in the FAC-batches.

Given the large similarities in plastic viscosity μ, a discussion about these data has been (implicitly) made in the previous section. However, the effect of plug flow and air entrainment on the plastic viscosity μ is not considered until in the next section and in Section 6.4.8.

Summary of Main Results

1. The **HMW Ca** polymer is more effective plasticizer than the **LMW Ca** polymer.

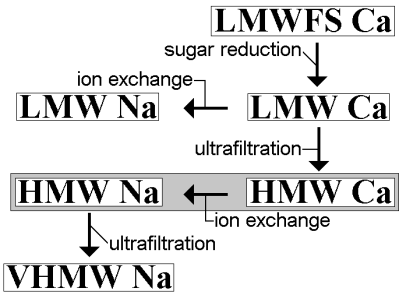
2. When using the **HMW Ca** polymer at 0.6% sbwc [$w/c = 0.4$; OPC], a clear decrease in yield value τ_o exists for the corresponding batches, with increasing temperature. Almost identical result is produced for the **HMW Na**-case, as shown in the previous section.

3. A possible explanation for this, is related to increased swelling and/or increased adsorption of the polymers.

4. As shown with Figure 6.27, the type of result shown in point 1. above, is not universal: When using the FAC at $w/c = 0.57$, a clear increase in yield value τ_o exists for the corresponding batches, with increasing temperature.

5. The time dependency of the shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ is generally more controlled by the yield value τ_o, rather than by the plastic viscosity μ (OPC). Some temperature dependency is present for the plastic viscosity μ (OPC).

6.4.5 HMW Na versus HMW Ca



As is shown in Table 4.2 (Page 74), there is not much difference between the two polymers **HMW Na** and **HMW Ca**. The sugar content and DOS values are almost the same. Also, the number average molecular weight M_{n} is exactly the same. However, the mass average molecular weight M_{m} is about 13% larger for the **HMW Na** polymer. With the few physical differences between the two polymer types, it is to expected that the viscometric values of the corresponding batches are more or less the same. As is shown with Figure 6.31, this condition applies when using the dosage of 0.3% sbwc. However, increasing this value up to 0.6% sbwc (and reducing the water content at the same time), some differences starts to occur for the mortar case. In all cases of 5°C and 23°C for mortar, the batches of the **HMW Ca** has always a lower yield value τ_{o}, relative to the **HMW Na**-batches. This is unexpected, given the similarities in the molecular weight values M_{n} and M_{m} of the two polymer types. When increasing the temperature from 23°C to 38°C, the situation is reversed. That is, the **HMW Na** starts to perform slightly better. This could be related to the larger mass average molecular weight M_{m} of the **HMW Na** polymer. With this, a slightly larger swelling and/or adsorption effect could be factual, with increasing temperature. See Section 6.4.4 about the swelling/adsorption theory and its relationship to the shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$.

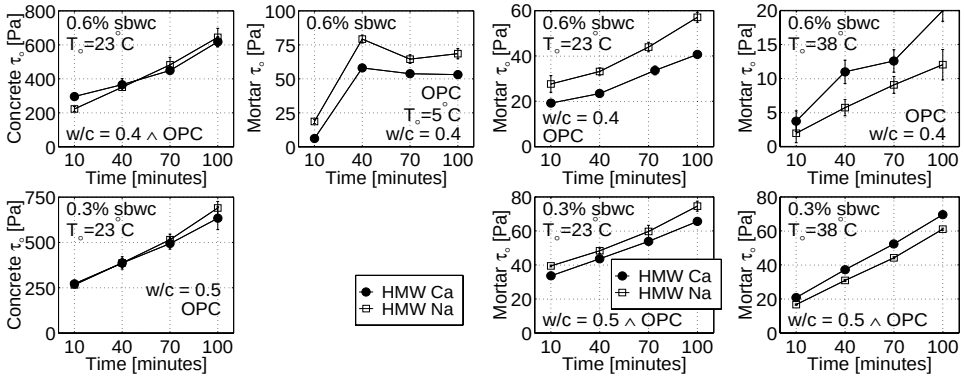


Figure 6.31: Measured yield value τ_{o} of concrete and mortar, using either the **HMW Na** or the **HMW Ca** polymer (OPC).

As is mentioned on Page 126, the mortar batches of [0.3% sbwc; $T_{\mathrm{o}} = 23°\mathrm{C} \wedge T_{\mathrm{o}} = 38°\mathrm{C}$; $w/c = 0.5$; OPC] have a large air stability. Also, the condition $\tau_{\mathrm{o}}/\mu \leq 100\,\mathrm{s}^{-1}$ applies mostly for this case. Hence, neither the plug flow nor the air entrainment will introduce a large error to the plastic viscosity μ for such circumstances. Therefore, the two bottom right illustrations of Figure 6.32 are considered to be relatively accurate. As is shown in these two illustrations, the plastic viscosity μ is more or less a constant with time for the case of mortar at 23°C, and is somewhat increasing with time for the 38°C-case [0.3% sbwc; $w/c = 0.5$; OPC]. Interestingly, the same behavior is measured for the case of mortar of [0.6% sbwc; $w/c = 0.4$; OPC]. The only difference between the two cases, is that the latter has roughly twice as large plastic viscosity μ, relative

to the former case. This overall similarity could indicate that the error from air entrainment and plug flow is not so large for the [0.6% sbwc; $w/c = 0.4$; OPC]-case presented here. It is also interesting to note that for either 0.6% sbwc or 0.3% sbwc, the plastic viscosity is always around order of magnitude smaller for the mortar case, relative to the concrete case. That is, at 0.6% sbwc, the concrete batch has a plastic viscosity of about $\mu = 20\,\mathrm{Pa}\cdot\mathrm{s}$, while for the mortar it is $2\,\mathrm{Pa}\cdot\mathrm{s}$. Likewise, at 0.3% sbwc, the values are $10\,\mathrm{Pa}\cdot\mathrm{s}$ and $1\,\mathrm{Pa}\cdot\mathrm{s}$, respectively.

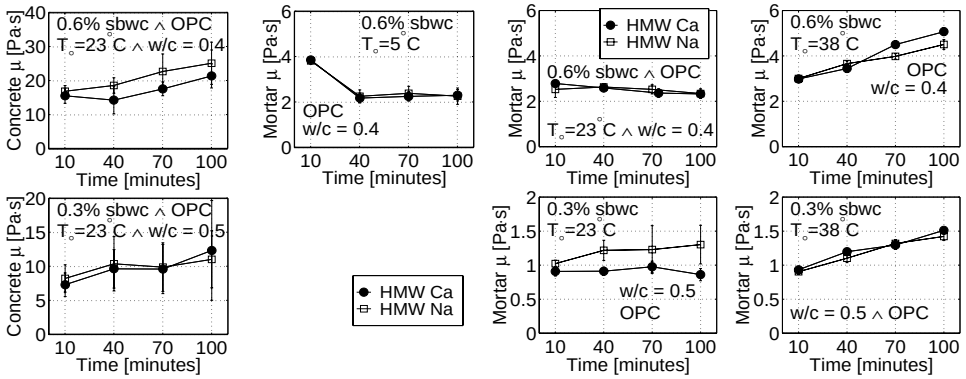


Figure 6.32: Measured plastic viscosity μ of concrete and mortar, using either the **HMW Na** or the **HMW Ca** polymer (OPC).

When comparing the concrete results with the corresponding mortar results (23°C) in Figure 6.31, one has to bear in mind the range of the error-bars $\pm\Delta\check{\tau}_\mathrm{o}$ (Equation 6.2) that apply for the fresh concrete. These error-bars are shown in Figures 6.2 and 6.8. As noted in these figures, the condition $\Delta\check{\tau}_\mathrm{o} \approx \Delta\tau_\mathrm{o}$ is mostly valid for the case of mortar, while for concrete, the condition $\Delta\check{\tau}_\mathrm{o} > \Delta\tau_\mathrm{o}$ is always valid. For further readings about these error-limits (or rather, confidence interval), see Section 6.2.2.

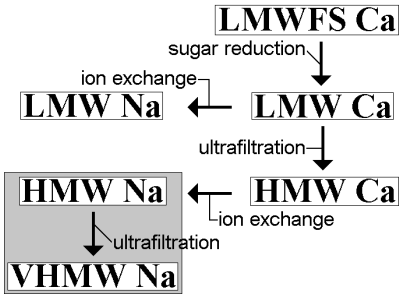
Summary of Main Results

1. Batches using either the **HMW Ca** or the **HMW Na** polymer at [$w/c = 0.5$; 0.3% sbwc] results roughly in the same set of viscometric values. Given the similarity between the two polymer types, this result is to be expected.

2. However, with [$w/c = 0.4$; 0.6% sbwc], some difference starts to emerge between the two polymer cases. At 5 and 23°C, the **HMW Ca** polymer gives somewhat better result, while at 38°C the **HMW Na** polymer makes a better performance.

3. The effect of the different polymer types is most often reflected by the measured yield value τ_o, rather than by the measured plastic viscosity μ. Also, there is a larger time and temperature dependency present for the yield value τ_o, relative to the plastic viscosity μ. Error from air entrainment and plug flow, does not seems to be much present for the plastic viscosity data shown in this section.

6.4.6 VHMW Na versus HMW Na



As is shown in Table 4.2 (Page 74), the main difference between the **VHMW Na** polymer and the **HMW Na** polymer, rests in the number average molecular weight and the mass average molecular weight M_{n} and M_{m}. As to be expected, the DOS value is somewhat smaller for the **VHMW Na** polymer, and its sugar content is almost nil (see Section 4.2.2). For the **HMW Na** polymer, the molecular weight values are $M_{\mathrm{n}} = 9900\,\mathrm{g/mol}$ and $M_{\mathrm{m}} = 84600\,\mathrm{g/mol}$. For the **VHMW Na** polymer, these values are high as $M_{\mathrm{n}} = 41600\,\mathrm{g/mol}$ and $M_{\mathrm{m}} = 183000\,\mathrm{g/mol}$. As is shown in Figure 4.3 (Page 75), this means that for the given dosage of 0.6% sbwc, a larger portion of the **VHMW Na** polymer is adsorbed on the surface of the cement particles, relative to the **HMW Na** polymer. With the larger adsorption and much larger dimension of the **VHMW Na** polymer, a considerable larger steric potential energy V_{S} is present between the cement particles using this polymer, compared to when using the **HMW Na** polymer. This gives a larger contribution to the energy barrier $V_{\mathrm{T}}^{\max}$ and increased reach $2/\kappa + 2D_{\mathrm{pol}} \approx 2D_{\mathrm{pol}}$ and therefore results in a smaller coagulation rate H (Equation 2.59). With this, a smaller coagulation state $J_{\mathrm{t}}^{\mathrm{tot}} \approx J_{\mathrm{t}}^{\mathrm{p}}$ is a reality (Equation 2.65), giving a lower shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$ (Sections 2.6.3). As should be apparent with the experimental results presented so far, the change in shear viscosity η is mainly reflected by the change in yield value τ_{o}. Hence, with the batches using the **VHMW Na** polymer, a lower yield value τ_{o} should be expected.

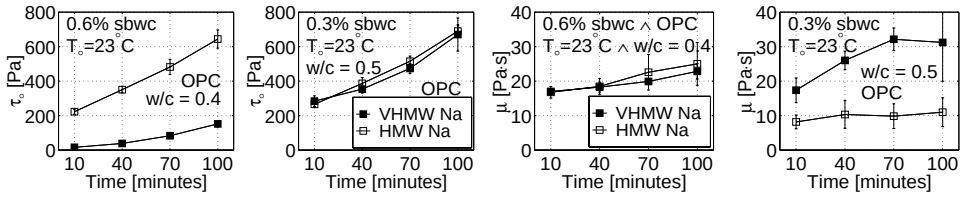


Figure 6.33: Measured yield value τ_{o} (to the left) and plastic viscosity μ (to the right) of concrete, using either the **HMW Na** or the **VHMW Na** polymer (OPC).

As Figure 6.33 (concrete) and Figure 6.34 (mortar) shows, a lower yield value τ_{o} is a reality for the batches using the **VHMW Na** polymer at the dosage of 0.6% sbwc (OPC, FAC). As discussed above, this behavior is to be expected. However, reducing the dosage down to 0.3% sbwc, approximately the same yield value τ_{o} is measured for both cases of **VHMW Na** and **HMW Na** (OPC, FAC). For the OPC-case, the water content is increased ($w/c = 0.4 \rightarrow 0.5$) at the same time as the polymer reduction (0.6% sbwc $\rightarrow$ 0.3%), and therefore this yield value behavior might be expected. More precisely, with the combination of high w/c-ratio and low polymer dosage, the special benefits of the **VHMW Na** polymer could be overshadowed by a too large quantity of water present in the corresponding OPC-batches (at $w/c = 0.5$). The same could apply for the FAC-case at 0.3% sbwc and $w/c = 0.57$ (see also the discussion in the end of Section 6.4.4).

As is mentioned in Section 6.4.5, neither the plug flow nor the air entrainment is considered to introduce a large error to the plastic viscosity μ for the mortar cases

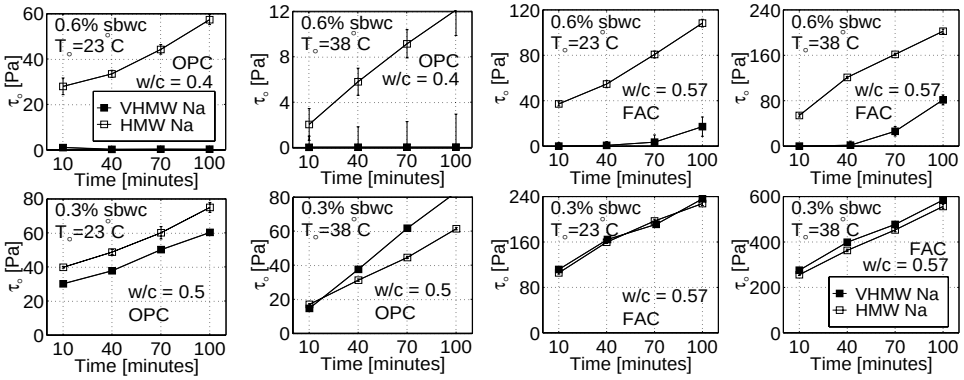


Figure 6.34: Measured yield value τ_o of mortar, using OPC and FAC with either the **HMW Na** or the **VHMW Na** polymer.

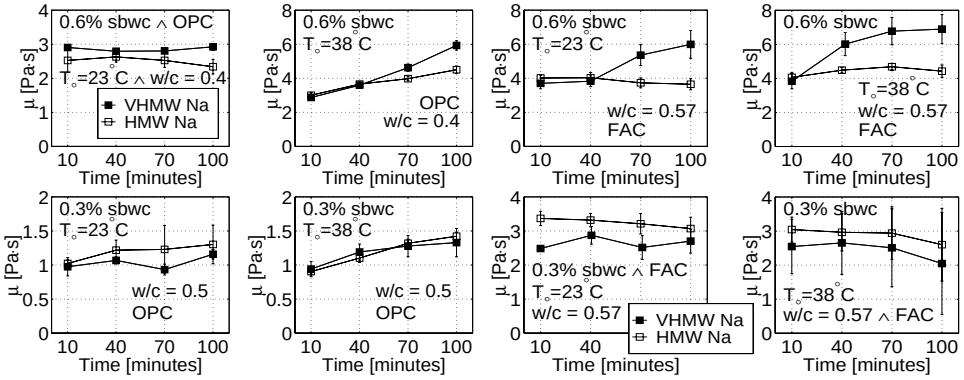


Figure 6.35: Measured plastic viscosity μ of mortar, using OPC and FAC with either the **HMW Na** or the **VHMW Na** polymer.

of [0.3% sbwc; $T_o = 23°C \wedge T_o = 38°C$; $w/c = 0.5$; OPC]. Therefore, the two bottom left illustrations of Figure 6.35 are considered to be relatively accurate. As shown in these two illustrations, the plastic viscosity μ is more or less a constant with time for the case of mortar at 23°C, and is somewhat increasing with time for the 38°C-case [0.3% sbwc; $w/c = 0.5$; OPC]. Interestingly, as shown in the two top left illustrations, the same behavior is measured for the case of mortar of [0.6% sbwc; $w/c = 0.4$; OPC]. The only difference between the two cases, is that the latter has a larger plastic viscosity values, relative to the former case. This overall similarity could indicate that the error from air entrainment and plug flow is not so large for the [0.6% sbwc; $w/c = 0.4$; OPC]-case presented here.

The FAC-batches show somewhat a similar plastic viscosity behavior, as for the OPC-batches. However, an exception to this is shown in the bottom right illustration of Figure 6.35 [0.3% sbwc; $T_o = 38°C$; $w/c = 0.57$; FAC]. There it appears that the plastic viscosity is somewhat decreasing with increasing time (but not significantly, c.f. the extent of the error-bars). As shown in the corresponding illustration of Figure 6.34, this specific case holds also the larger yield values. Relative to this, it is interesting to bear in mind the text written on Page 113.

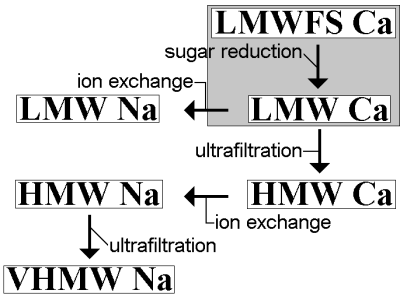
Generally, for all plastic viscosity results μ shown in Figure 6.35, there is not much significant difference present between the two different polymer cases. When significant difference occurs, it is mostly the **VHMW Na**-batches that gives a larger plastic viscosity μ. Also, quite frequently, little temperature and time dependency is present for the plastic viscosity μ.

Summary of Main Results

1. At the dosage of 0.3% sbwc, the **VHMW Na** polymer performs equally as well as the **HMW Na** polymer; i.e. the same or similar set of viscometric values τ_o and μ is measured in either case of **VHMW Na** or **HMW Na**.

2. At higher dosage of 0.6% sbwc, the **VHMW Na** polymer starts to perform far better than the **HMW Na** polymer. More precisely, different yield values τ_o are measured for the two cases, while the plastic viscosity values μ are more or less the same.

6.4.7 LMWFS Ca versus LMW Ca



When comparing the **LMWFS Ca** polymer with the **LMW Ca** polymer in Table 4.2, it is apparent that the **LMW Ca** consist of larger molecular weight values M_{n} and M_{m} by about 9% and 33%, respectively. DOS values are about the same for both polymer types. With the importance of the molecular weight towards the steric hindrance V_{S}, it is to be expected that the **LMWFS Ca**-batches will have a higher yield value τ_{o} (i.e. larger shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$) relative to the **LMW Ca**-batches. As shown in Figures 6.36 and 6.37, this is measured for most cases. However, when dosage of 0.6% sbwc is used, the situation is somewhat reversed at 10 minutes after water addition. That is, the yield value τ_{o} at 10 minutes is slightly lower for the **LMWFS Ca**-batches. This might be related to the larger sugar content present in the **LMWFS Ca** polymer (22% versus 7%), which results in slower hydration. The point where the **LMW Ca**- and **LMWFS Ca**-batches have roughly the same yield value τ_{o}, is marked with a small box in the corresponding illustrations.

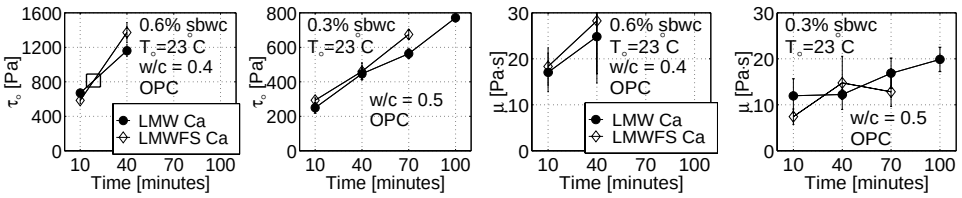


Figure 6.36: Measured yield value τ_{o} (to the left) and plastic viscosity μ (to the right) of concrete, using either the **LMW Ca** or the **LMWFS Ca** polymer (OPC).

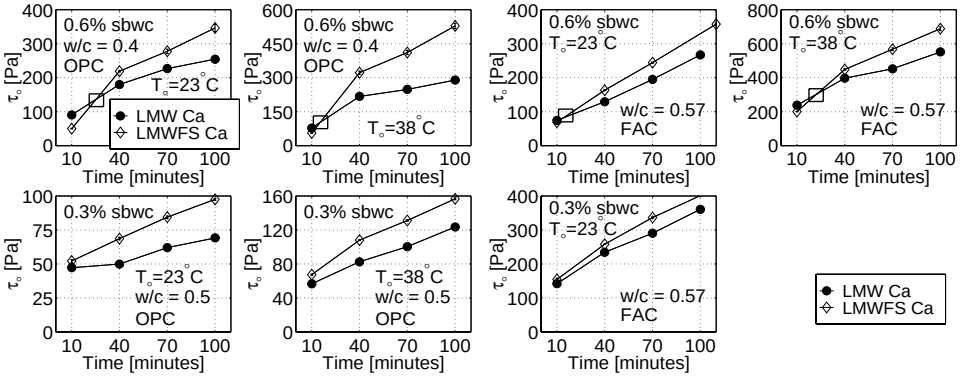


Figure 6.37: Measured yield value τ_{o} of mortar, using OPC and FAC with either the **LMW Ca** or the **LMWFS Ca** polymer.

Often, around and beyond 40 minutes, the difference in yield value τ_{o} between the cases of **LMW Ca** and **LMWFS Ca** are somewhat larger than expected relative to the (moderately small) difference in molecular weight values M_{n} and M_{m}. The reason for this large difference is unclear. The large difference is in particular apparent for

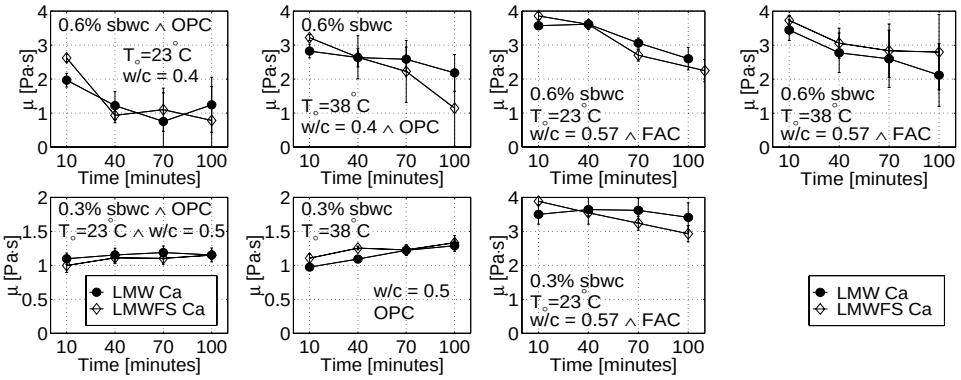


Figure 6.38: Measured plastic viscosity μ of mortar, using OPC and FAC with either the **LMW Ca** or the **LMWFS Ca** polymer.

the cases of [0.6% sbwc; $T_o = 38°C$; $w/c = 0.4$; OPC] and [0.3% sbwc; $T_o = 23°C$; $w/c = 0.5$; OPC].

As is mentioned in Section 6.4.5, neither the plug flow nor the air entrainment is considered to introduce a large error to the plastic viscosity μ for the mortar cases of [0.3% sbwc; $T_o = 23°C \wedge T_o = 38°C$; $w/c = 0.5$; OPC]. Therefore, the two bottom left illustrations of Figure 6.38 are considered to be relatively accurate. As shown in these two illustrations, the plastic viscosity μ is more or less a constant with time for the case of mortar at 23°C, and is slightly increasing with time for the 38°C-case [0.3% sbwc; $w/c = 0.5$; OPC]. This behavior is not reproduced for mortar of [0.6% sbwc; $w/c = 0.4$; OPC], as is apparent with the two top left illustrations. There, it is rather shown that with increasing time, the plastic viscosity μ is decreasing. On Page 113, large increase in yield value τ_o, is related to decrease in plastic viscosity μ, through air entrainment; i.e. the drop in plastic viscosity shown here, could be related to error generated by air entrainment. A drop in plastic viscosity μ as a function of time is also observed for the FAC-case, in the right illustration of Figure 6.38.

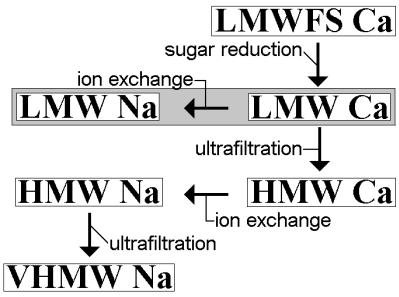
Generally, for Figure 6.36 (concrete) and Figure 6.38 (mortar), there is no significant difference between the measured plastic viscosity μ of the batches using the two polymer types **LMWFS Ca** and **LMW Ca**. Their difference rests rather in the yield value τ_o.

Summary of Main Results

1. Generally, the yield value τ_o of the **LMWFS Ca**-batches is larger than of the **LMW Ca**-batches. This is not unexpected, given the difference in molecular weight numbers M_n and M_m between the two polymer types.

2. However, in some cases, the difference seems to be larger than can be explained by the molecular weight values alone.

6.4.8 LMW Na versus LMW Ca



The main difference between the **LMW Na** and **LMW Ca** is that the former has gone under an ion exchange while the latter has not (see Figure 4.2). The interesting difference between the two polymer types rests in the molecular weight values M_{n} and M_{m}. For the **LMW Na** polymer, the molecular weight values are $M_{\mathrm{n}} = 7100\,\mathrm{g/mol}$ and $M_{\mathrm{m}} = 60900\,\mathrm{g/mol}$. For the **LMW Ca** polymer, these values are slightly higher and consist of $M_{\mathrm{n}} = 7400\,\mathrm{g/mol}$ and $M_{\mathrm{m}} = 73100\,\mathrm{g/mol}$. Hence, one could expect a slightly larger steric potential energy V_{S} for the **LMW Ca**-case, resulting in a slightly larger contribution to the energy barrier $V_{\mathrm{T}}^{\max}$ and reach $2D_{\mathrm{pol}}$ (c.f. $V_{\mathrm{T}} = V_{\mathrm{R}} + V_{\mathrm{A}} + V_{\mathrm{S}}$). If so, this would give a smaller coagulation rate H (see Equation 2.59, Page 46). With a smaller coagulation rate H, a smaller coagulation state $J_{\mathrm{t}}^{\mathrm{tot}} \approx J_{\mathrm{t}}^{\mathrm{p}}$ results (c.f. Equation 2.65, Page 50), which again gives a lower shear viscosity η (see Section 2.6.3). With this, then at first consideration, one could expected that the **LMW Ca**-case should have a slightly lower shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$, relative to the **LMW Na**-case (then reflected with a slightly lower yield value τ_{o}). Except for the case of [0.3% sbwc; $T_{\mathrm{o}} = 23°\mathrm{C}$; $w/c = 0.5$; OPC], the opposite is measured. That is, the **LMW Na**-case gives most often a lower yield value τ_{o}. This is apparent with Figures 6.39 and 6.40.

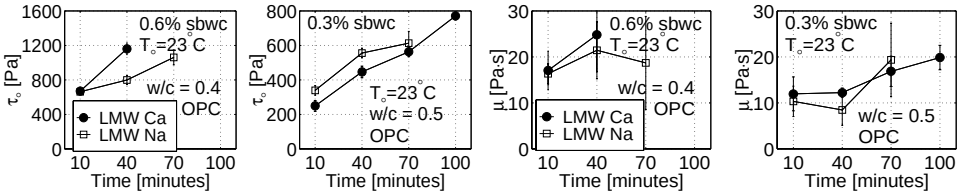


Figure 6.39: Measured yield value τ_{o} (to the left) and plastic viscosity μ (to the right) of concrete, using either the **LMW Na** or the **LMW Ca** polymer (OPC).

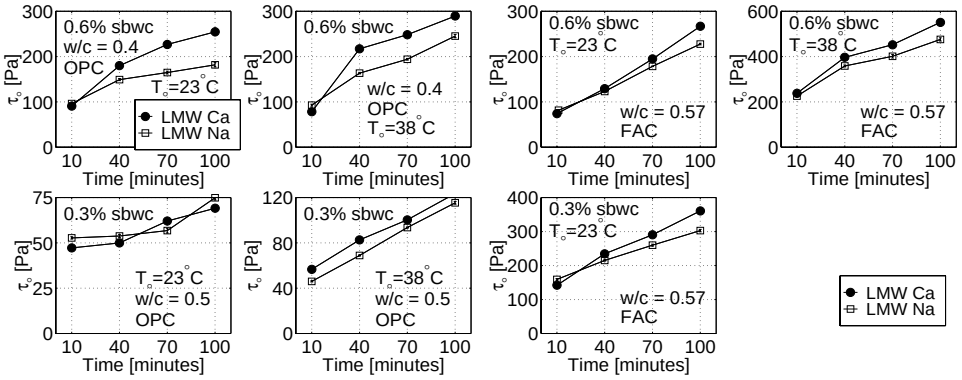


Figure 6.40: Measured yield value τ_{o} of mortar, using OPC and FAC with either the **LMW Na** or the **LMW Ca** polymer.

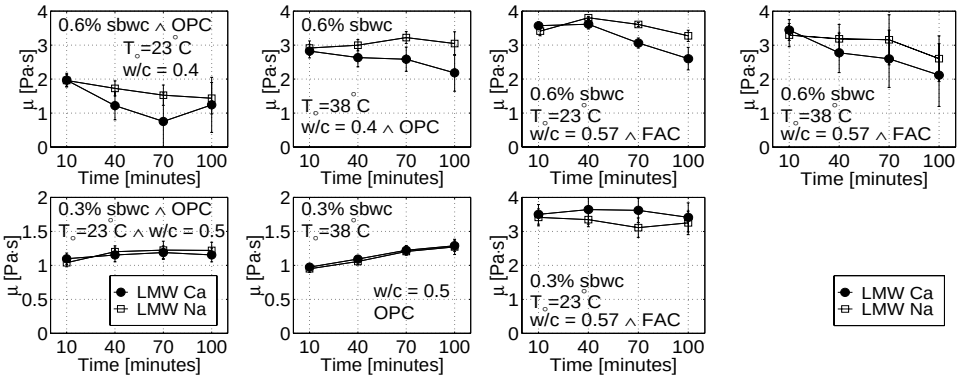


Figure 6.41: Measured plastic viscosity μ of mortar, using OPC and FAC with either the **LMW Na** or the **LMW Ca** polymer.

The type of difference between the **LMW Na**- and **LMW Ca**-cases, shown in Figures 6.39 and 6.40, has been reported elsewhere [105, 128]. This difference can perhaps rather be explained through electrostatic considerations (V_{R}): With an (active) cation of valence $z_{\mathrm{Ca}} = 2$ (**LMW Ca** $\rightarrow$ **LMW**$^{-2n}$ + n Ca^{+2}), a larger contribution to the ionic strength I is likely. This is more apparent by examining Equation 2.54 (Page 40) in a more detail. With a valence of $z_{\mathrm{i}} = z_{\mathrm{Ca}} = 2$, the calcium ion make a four times larger contribution to the ionic strength ($z_{\mathrm{i}}^2 = 4$), relative to the sodium ion with $z_{\mathrm{i}} = z_{\mathrm{Na}} = 1$ (**LMW Na** $\rightarrow$ **LMW**$^{-\mathrm{n}}$ + n Na^{+}). As shown in the left illustration of Figure 2.19 (Page 42), increased ionic strength I gives a reduced energy barrier $V_{\mathrm{T}}^{\max}$ and a smaller reach $2/\kappa$. This results in a larger coagulation rate H and hence in an increased coagulation state $J_{\mathrm{t}}^{\mathrm{tot}}$, which is then observed with a larger shear viscosity $\eta = \mu + \tau_{\mathrm{o}}/\dot{\gamma}$ (reflected here by a larger yield value τ_{o} shown in Figures 6.39 and 6.40).

From the above, the electrostatic potential energy V_{R} seem to be important, at least for the low molecular weight polymers like of the **LMW Na** and **LMW Ca**. As shown in Figure 4.3 (Page 75), smaller lignosulfonates gives a smaller adsorption. With a smaller adsorption and a smaller dimension of the low molecular weight (**LMW**) lignosulfonates, a smaller reach $2D_{\mathrm{pol}}$ and magnitude of the steric potential energy V_{S} results. With a diminished steric potential energy V_{S}, the electrostatic contribution V_{R} becomes more important for the final outcome.

The rheological behavior of all the high molecular weight (**HMW**) batches shown previously, has been successfully explained through the molecular weight values M_{n} and M_{m}. This indicates that the steric potential energy V_{S} is more or less overshadowing the electrostatic potential energy V_{R} for such cases. As such, issues like ionic strength I or zeta potential ζ becomes less important.

As is mentioned in Section 6.4.5, the plastic viscosity μ for mortar cases of [0.3% sbwc; $T_{\mathrm{o}} = 23°\mathrm{C} \wedge T_{\mathrm{o}} = 38°\mathrm{C}$; $w/c = 0.5$; OPC] are considered to be relatively accurate. The corresponding results are shown in the two bottom left illustrations of Figure 6.41. As shown there, the plastic viscosity μ is more or less a constant with time, for the case of mortar at 23°C, and is slightly increasing with time for the 38°C-case [0.3% sbwc; $w/c = 0.5$; OPC]. This behavior is not so well reproduced for mortar of [0.6% sbwc; $w/c = 0.4$; OPC], as is apparent with the two top left illustrations. Rather, there is some indication of drop in plastic viscosity μ with increasing time.

As mentioned in the previous section, this drop is related to increased air content with time. This drop is more apparent for the FAC-case shown with the two top right illustrations in Figure 6.41.

Generally, for Figure 6.39 (concrete) and Figure 6.41 (mortar), there is no significant difference between the measured plastic viscosity μ of the batches using the two polymer types **LMW Na** and **LMW Ca**. Their difference rests rather in the yield value τ_o.

Summary of Main Results

1. The **LMW Na** polymer is somewhat more effective plasticizer than the **LMW Ca** polymer.

2. The electrostatic potential energy V_R seems to becomes more important for the low molecular weight lignosulfonates, making issues like ionic strength I and/or zeta potential ζ more relevant.

6.4.9 SNF versus HMW Na

In the previous sections, the effect of different lignosulfonate types is compared internally. All of them are produced by BORREGAARD LIGNOTECH. It is now interesting to compare for example, the **HMW Na** polymer with a "outside" product, like of the Sulfonated Naphthalene Formaldehyde (**SNF**). As mentioned in Chapter 4, the **SNF** polymer has the commercial name SUPAREX M40 and is produced by HODGSON CHEMICALS LTD. Unfortunately, no (compatible) molecular weights values M_n or M_m where available for this product, at the time of writing. This limits the discussion somewhat. The comparison between the batches using the **SNF** and the **HMW Na** polymer, is shown in Figure 6.42 to Figure 6.44.

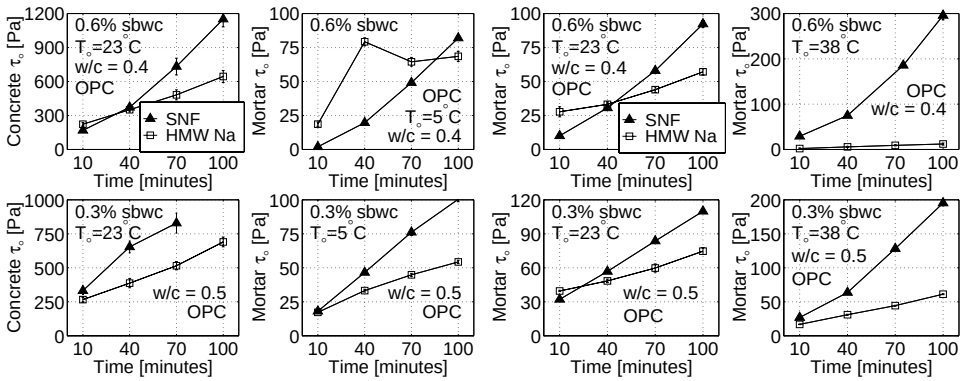


Figure 6.42: Measured yield value τ_o of concrete and mortar, using either the **SNF** or the **HMW Na** polymer (OPC).

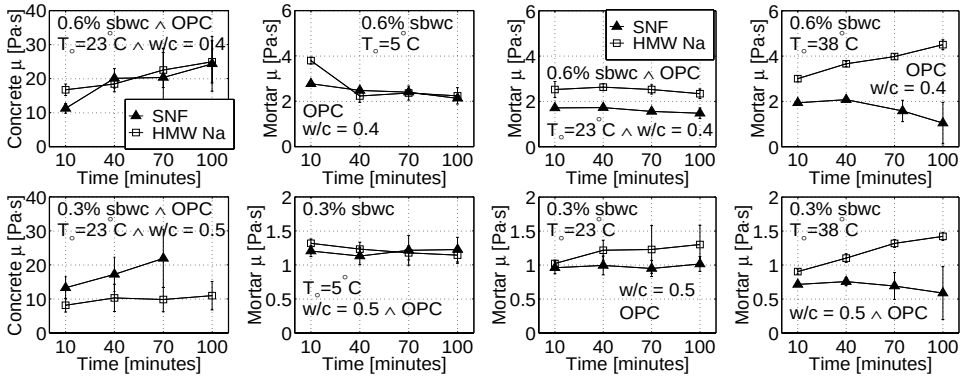


Figure 6.43: Measured plastic viscosity μ of concrete and mortar, using either the **SNF** or the **HMW Na** polymer (OPC).

In almost all cases, the **SNF**-batches have a faster increase in yield value τ_o with time, relative the **HMW Na**-batches. This applies regardless of concrete, mortar, w/c-ratio, cement type or temperature conditions. With little time evolution in plastic viscosity μ (Figures 6.43 and 6.44), one could state that the workability retention for the **SNF**-case is worse, relative to the **HMW Na**-case. This is perhaps more apparent when observing the slump results shown in Figures 6.5 and 6.11 (Page 108).

For the case of [0.6% sbwc; w/c = 0.4; OPC] in Figure 6.42, it is interesting to note how the **HMW Na**-case and the **SNF**-case go their separate ways, with increasing temperature. The for the **SNF**-case, there is a rapid increase in yield value τ_o, with increasing temperature. However, for the **HMW Na**-case the reverse trend is observed (see also Section 6.4.3). A similar, but not so as clear, tendency is observed for the case of [0.3% sbwc; $w/c = 0.5$; OPC] in the same figure.

In Figures 6.42 and 6.44, it is interesting to note that the yield value τ_o is in most cases lower for the **HMW Na**-batches at the dosage of 0.3% sbwc (OPC, FAC). At the dosage of 0.6% sbwc, the **SNF**-batches has often equal or better initial performance (i.e. lower yield value τ_o), but always a worse final performance at 100 minutes. This is because of the pre-mentioned faster increase in yield value τ_o as a function of time.

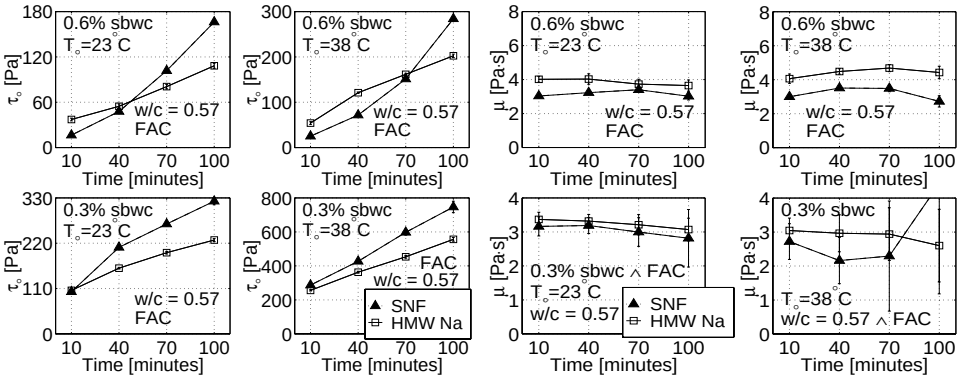


Figure 6.44: Measured yield value τ_o (to the left) and plastic viscosity μ (to the right) of mortar, using either the **SNF** or the **HMW Na** polymer (FAC).

From Figures 6.43 and 6.44, there is seldom any significant difference between the measured plastic viscosity μ of the batches using the two polymer types. In the few cases when significant difference is present, it is the **SNF**-batches that have a lower plastic viscosity μ. This difference could rather be related to error generated by air entrainment for the **SNF**-case, rather than being a particular properties of the **SNF** polymer (see Page 113).

Summary of Main Results

1. Similar rheological valuer are roughly measured at 10 minutes, using either the **SNF** polymer or the **HMW Na** polymer. However, for the **SNF**-case there is a larger increase in yield value τ_o as a function of time. That is, a larger "workability loss" or "slump loss" is present for the **SNF**-case.

2. The effect of the two polymer types **SNF** and **HMW Na**, is more strongly reflected by the yield value τ_o, than by the plastic viscosity μ.

6.5 Summary and Conclusion

6.5.1 Section 6.2: The Overall Results

The change in shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ as a function of time and temperature, is more reflected by the change in yield value τ_o, than by the change in plastic viscosity μ; i.e. the yield value τ_o is more sensitive to temperature and time, relative to the plastic viscosity μ (see also the numbered items on Page 126).

The effect of the different polymer types is more reflected by the measured yield value τ_o, than by the measured plastic viscosity μ; i.e. comparing the effect of two different polymer types gives usually a different yield value τ_o, while the plastic viscosity μ is roughly the same for the two cases.

6.5.2 Section 6.3: The Concrete Results

With Equation 6.6 (Page 122), a certain relationship between the Abrams slump cone and the ConTec **BML** Viscometer 3 is established. This relationship is further enhanced by Equation 6.7. The right illustration of Figure 6.22 (Page 123) demonstrates the use of the last-mentioned equation. With $R^2 = 0.86$ between the calculated slump by Equation 6.7 and the actual measured slump, a strong relationship exits between the ConTec **BML** Viscometer 3 and the Abrams slump cone. This applies even though the problem of gravel migration (Chapter 10) is occurring for the concrete batches of this thesis.

As mentioned in Section 4.3.2, the mix design of the OPC-mortars used in this thesis, are calculated directly from the mix design of the corresponding concrete mixes. As such, it is to be expected that some kind of relationship exists between their measured viscometric values. Figures 6.23 and 6.24 (Page 124) shows the degree of such relationship. These figures demonstrate a strong relationship between the yield values of concrete and mortar, and a weak relationship between the corresponding plastic viscosity values. This poor correlation in plastic viscosity, is most likely somewhat related to the error generated by plug flow (see Section 3.5.3) and by air entrainment (see Section 5.5.3) in the mortar case. As has been previously mentioned, the plastic viscosity μ is much more sensitive to these errors, relative to the yield value τ_o. Hence a larger scatter in data results in Figure 6.24, relative to Figure 6.23. The gravel migration in concrete (Chapter 10) is probably also somewhat responsible for the scatter shown in Figures 6.23 and 6.24.

6.5.3 Section 6.4: The Mortar Results

Generally, the yield value τ_o is measured to be increasing with increasing time. Most often, it is also increasing with increasing temperature. However, as mentioned in Sections 6.4.3 and 6.4.4, there are exceptions to this: For mortar samples (OPC) using the **HMW Na** and **HMW Ca** polymer at 0.6% sbwc, there is a clear decrease in yield value τ_o with increasing temperature. A possible explanation for this, is related to increased swelling and/or increased adsorption of the **HMW Na** and **HMW Ca** polymers (see Figure 6.30, Page 138). The same characteristics could be expected when using the **VHMW Na** polymer in mortar of [0.6% sbwc; OPC]. However, since the yield value for such case is regardless always measured zero, no such effect could be registered. This temperature effect does not occur for the case of FAC. The reason for this may possible be related to that the swelling/adsorption properties of

these polymers are overshadowed by a too large quantity of water (relative to cement content) present in the FAC-batches (large fineness (Blaine) of the FAC could also play a role here; see Section 4.2.1).

Reliable plastic viscosity data, gives that μ is either a constant with time, or is slightly increasing with time. Occasionally, the plastic viscosity μ is also decreasing with increasing time. This last-mentioned behavior is rather related to error generated by air entrainment, than being considered as an actual physical property of the polymer used in the corresponding mortar batch (see discussion on Page 113).

The rheological behavior of mortar and concrete, using the high molecular weight lignosulfonates (**VHMW** and **HMW**), are successfully explained through the molecular weight values M_{n} and M_{m}. This indicates that the steric potential energy V_{S} is more or less overshadowing the electrostatic potential energy V_{R} for such cases. As such, issues like ionic strength I or zeta potential ζ becomes less important. However, the electrostatic potential energy V_{R} seems to become more important for the batches using the low molecular weight lignosulfonates (**LMW**), making issues like ionic strength I and zeta potential ζ more relevant.

At the dosage of 0.6% sbwc, the **VHMW Na** polymer starts to perform far better than the **HMW Na** polymer. More precisely, different yield values τ_{o} are measured for the two cases, while the plastic viscosity values μ are more or less the same. This is not unexpected given their difference in the molecular weight values M_{n} and M_{m}. However, at the dosage of 0.3% sbwc, the **VHMW Na** polymer performs only equally as well as the **HMW Na** polymer; i.e. the same or similar set of viscometric values τ_{o} and μ are measured in either case of **VHMW Na** or **HMW Na**. This unexpected result applies for both concrete and mortar, regardless of temperature, w/c-ratio or cement type (OPC, FAC).

A faster increase in yield value τ_{o} with time, is present in the mortar batches when using the **SNF** polymer, relative when using the **HMW Na** polymer (OPC, FAC). This difference in workability loss is also observed by the slump data of concrete, as shown in Figures 6.5 and 6.11 (Pages 108 and 114). This demonstrates the overall good performance of the natural lignosulfonate product, over the synthetic produced sulfonated naphthalene formaldehyde (or **SNF**). In fact, the (natural) **VHMW Na**, **HMW Na** and **HMW Ca** polymers, demonstrate a far better performance in terms of workability and workability retention, relative to the (synthetic) **SNF** polymer. This is quite apparent with Figures 6.1 (Page 104), 6.7 (Page 110) and 6.17 (Page 118).

6.5.4 Rating of Polymer types

To conclude in a simple manner, the rating of the different polymer types from the best type to the worst type is as follows: **1**) **VHMW Na** (the best type); **2**) **HMW Ca**; **3**) **HMW Na**; **4**) **SNF**; **5**) **LMW Na**; **6**) **HMW Ca**; **7**) **LMWFS Ca** (worst).

6.5.5 Recommended Future Research

What is missing in this work, is the information about ionic strength I and zeta potential ζ for each batch. With such information, a better evaluation of the electrostatic repulsive potential energy V_{R} would be present, relative to the steric potential energy V_{S}, for the different polymer cases. If a similar research program to this is to be conducted, it is recommended that in addition to the viscometric values τ_{o} and μ, the I and ζ values should be measured in parallel. Of course information about the molecular weight values of the polymers used should also be present.

The **VHMW Na** product is a new and an experimental product by BORREGAARD LIGNOTECH. At the dosage of 0.6% sbwc, its performance becomes quite remarkable. However, at the dosage of 0.3% sbwc, it only performs similar to the **HMW Na** and **HMW Ca** products. It would be very interesting to compare the **VHMW Na** and **HMW Na** products at a tighter dosage interval, for example at 0.1%, 0.2%, 0.3%, 0.4%, 0.5% and 0.6% sbwc (and keeping other variables constant at the same time, like the amount of cement and w/c-ratio). Such a test would give a clearer picture of the degree of similarities and dissimilarities between the two products **HMW Na** and **VHMW Na**. Including a 3rd generation of plasticizers, consisting of grafted carboxylic synthetic polymers, into such an investigation would also be interesting. It is recommended that not too high w/c-ratio is applied in such investigation. As previously shown in Section 6.2.4, when applying the w/c-ratio of 0.6 with the low dosage 0.1% sbwc, the effect of the polymers becomes overshadowed by a too large quantity of water.

Chapter 7

Computational Rheology

7.1 Introduction

In Chapters 8 and 9, a numerical simulation of the ConTec **BML** Viscometer 3 and ConTec Viscometer 4 is presented. The fluid consists of a viscoplastic material, flowing either under steady state condition or under transient condition. The objectives of the simulations are described in the introductory part of each chapter. The overall theory presented in the current chapter, will be used in Chapters 8, 9 and 10 when simulating viscoplastic flow inside six different viscometers, listed below:

- ConTec **BML** Viscometer 3 (Chapter 8)
- ConTec Viscometer 4 (Figure 7.1, and Chapters 8 and 9)
- BTRHEOM (Chapter 10)
- A series of prototype viscometers presented in Chapter 10.

Simulation of a viscoplastic material is a challenging task. One reason for this, is that such material can be in a two different states, namely the viscoplastic state and the solid state. In the current work, a combination of several different techniques is necessary to produce solution of the corresponding nonlinear **p**artial **d**ifferential **e**quation (**PDE**). The techniques used here are the **Successive Substitution**, the **Continuation Method** (for steady state only) and the incorporation of the **Regularization Parameter**. Also, to save storage space and to be able to utilize the **Thomas Algorithm**, the **Alternating Direction Implicit** (**ADI**) technique is used. The method of **Finite Differences** is used when converting the PDE into its algebraic counterpart.

7.2 Governing Equation

Going through the same steps as was done in Section 3.3, but now assuming a height dependence z, gives the velocity profile described with Equation 7.1.

$$\mathbf{v} = v_\theta(r, z, t)\, \mathbf{i}_\theta \tag{7.1}$$

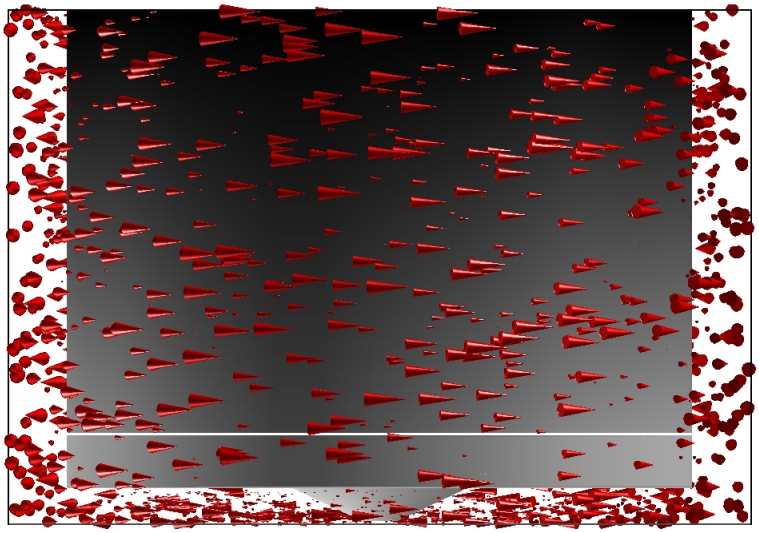


Figure 7.1: Velocity vector field $\mathbf{v}$ for a thixotropic fluid inside the ConTec Viscometer 4. The particular result shown applies at $t = 11\,\mathrm{s}$ after start of a measurement and is reproduced from Figure 9.5 on Page 217 (**VHMW Na** $\wedge\ t = 42\,\mathrm{min}$).

From Equation 7.1 the velocity gradient tensor $\nabla\mathbf{v}$ can be calculated, which through Equation 2.20 (Page 17), gives the strain rate tensor:

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right)(\mathbf{i}_\mathrm{r}\mathbf{i}_\theta + \mathbf{i}_\theta\mathbf{i}_\mathrm{r}) + \frac{1}{2}\frac{\partial v_\theta(r,z,t)}{\partial z}(\mathbf{i}_\mathrm{z}\mathbf{i}_\theta + \mathbf{i}_\theta\mathbf{i}_\mathrm{z}) \tag{7.2}$$

Substituting the above result in Equation 2.24 gives, after some derivations, the shear rate that applies inside the viscometer:

$$\dot{\gamma} = \sqrt{\left[\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right]^2 + \left[\frac{\partial v_\theta(r,z,t)}{\partial z}\right]^2} \tag{7.3}$$

According to Equations 2.23 and 7.2, the extra stress tensor $\mathbf{T}$ becomes:

$$\begin{aligned}\mathbf{T} = {} & \eta(r,z,t)\left(\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right)(\mathbf{i}_\mathrm{r}\mathbf{i}_\theta + \mathbf{i}_\theta\mathbf{i}_\mathrm{r}) + \\ & + \eta(r,z,t)\,\frac{\partial v_\theta(r,z,t)}{\partial z}(\mathbf{i}_\mathrm{z}\mathbf{i}_\theta + \mathbf{i}_\theta\mathbf{i}_\mathrm{z})\end{aligned} \tag{7.4}$$

The von Mises shear stress τ (see previous discussion in Section 3.2.1) can either be calculated with the help from Equation 7.4 or from Equation 7.3. This is apparent since $\tau^2 = -\mathrm{II}_\mathbf{S}^\mathrm{P} = (\mathbf{T}:\mathbf{T})/2 = (2\,\eta)^2\,\dot{\boldsymbol{\varepsilon}}:\dot{\boldsymbol{\varepsilon}}/2 = \eta^2\,(2\,\dot{\boldsymbol{\varepsilon}}:\dot{\boldsymbol{\varepsilon}}) = (\eta\,\dot{\gamma})^2$. Both approaches produce Equation 7.5.

$$\tau = \eta(\dot{\gamma},t)\left(\left[\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right]^2 + \left[\frac{\partial v_\theta(r,z,t)}{\partial z}\right]^2\right)^{\frac{1}{2}} \tag{7.5}$$

By using Equations 2.23 and 7.4 in Equation 2.17 (Page 16) with the concomitant use of the velocity profile described with Equation 7.1, the following governing equation

is produced:

$$\rho\frac{\partial v_\theta(r,z,t)}{\partial t} = \frac{\partial T_{r\theta}(r,z,t)}{\partial r} + 2\frac{T_{r\theta}(r,z,t)}{r} + \frac{\partial T_{z\theta}(r,z,t)}{\partial z} \tag{7.6}$$

The above equation applies in the θ-direction. In this derivation, two other equations are also produced: $0 = -\partial p(r,z,t)/\partial z - \rho\, g$ and $-\rho v_\theta^2(r,z,t)/r = -\partial p(r,z,t)/\partial r$. These apply in z- and r-directions, respectively. Fortunately, they are not directly coupled to Equation 7.6 and hence need not to be included in the numerical simulation. The terms $T_{r\theta}$ and $T_{z\theta}$ in the above are extracted directly from Equation 7.4:

$$T_{r\theta}(r,z,t) = \eta(r,z,t)\left(\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right) \tag{7.7}$$

$$T_{z\theta}(r,z,t) = \eta(r,z,t)\frac{\partial v_\theta(r,z,t)}{\partial z} \tag{7.8}$$

When discretizing the governing Equation 7.6, one might be tempted to first put Equations 7.7 and 7.8 into this equation and then expanding the velocity terms, for example by converting the term $\partial[\eta\,\partial v_\theta/\partial z]/\partial z$ into $\partial v_\theta/\partial z\ \partial\eta/\partial z + \eta\,\partial^2 v_\theta/\partial z^2$. According to Langtangen [66], such a procedure should be avoided when the shear viscosity η is a variable and not a constant. Therefore a discretization of Equation 7.6 as it is, will be done. The method to be used when converting Equation 7.6 into system of algebraic equations $\mathbf{A}\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$, is known as the method of **finite differences** [2, 66, 33]. The term k represents the **time index** in such way that $t = \mathrm{k}\,\Delta t$ represents the time and Δt the time step.

7.3 Method of Finite Differences

7.3.1 Introduction

For the sake of continuity of the text, a short introduction to the method of finite differences is given here. The general quest is to replace the relevant PDE with a system of algebraic equations. In the method of finite differences, this means converting the differential $\partial f/\partial x$ into its algebraic counterpart with the aid of Taylor series.

Taylor Approximation

The Taylor counterpart of a function $f(x)$ at the point $x = x_0 + \Delta x$ is given by [30]:

$$f(x_0+\Delta x) = \sum_{\mathrm{k}=0}^{\mathrm{n}} \frac{f^{(\mathrm{k})}(x)|_{x_0}}{k!}\Delta x^{\mathrm{k}} + R_{\mathrm{n}}(x_0+\Delta x) \tag{7.9}$$

where $f^{(\mathrm{k})}(x)|_{x_0}$ represents the n^{th} derivative of the function f at the point x_0. The term $R_{\mathrm{n}}(x)$ is called the $\mathbf{n}^{\mathrm{th}}$ **degree remainder** [30] for the function f. It is the error produced when replacing $f(x_0+\Delta x)$ with its approximation $\sum f^{(\mathrm{k})}(x)|_{x_0}\Delta x^{\mathrm{k}}/k!$. In other words, $R_{\mathrm{n}}(x_0+\Delta x) = f(x_0+\Delta x) - \sum f^{(\mathrm{k})}(x)|_{x_0}\Delta x^{\mathrm{k}}/k!$ represents the **truncation error**. An overestimate of the remainder is given by [30]:

$$|R_{\mathrm{n}}(x_0+\Delta x)| \leq \frac{\left|f^{(\mathrm{n}+1)}(x)|_z\right|\,\Delta x^{\mathrm{n}+1}}{(n+1)!} \tag{7.10}$$

where $z \in [x_0, x_0 + \Delta x]$ is chosen in such a way that the term $\left|f^{(n+1)}(x)|_z\right|$ has the largest value possible. Unfortunately, this value is not known and therefore it is only possible to claim that the magnitude of error varies as Δx^{n+1}. Hence, truncating the series after the n^{th} term, yields the (truncation) error $R_n(x_0 + \Delta x) = O(\Delta x^{n+1}) \propto \Delta x^{n+1}$.

Discretization

Discretization is the name of the process in converting the PDE into a system of algebraic equations. For the method of finite differences, this means converting the differential $\partial f/\partial x$ into its algebraic counterpart with the aid of Taylor series. Here, the function f is assumed to be dependent on x, y, z, and t. However, only two variables, namely x or t, are shown in the derivations that now follows. From Equation 7.9, the 2^{nd} degree Taylor polynomials of the function $f(x)$ at the spatial points $x_0 + \Delta x/2$ and $x_0 - \Delta x/2$ are given by Equations 7.11 and 7.12.

$$f(x_0 + \frac{\Delta x}{2}) = f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x_0} \frac{\Delta x}{2} + \frac{1}{2} \left.\frac{\partial^2 f}{\partial x^2}\right|_{x_0} \left[\frac{\Delta x}{2}\right]^2 + O^+(\Delta x^3/8) \tag{7.11}$$

$$f(x_0 - \frac{\Delta x}{2}) = f(x_0) - \left.\frac{\partial f}{\partial x}\right|_{x_0} \frac{\Delta x}{2} + \frac{1}{2} \left.\frac{\partial^2 f}{\partial x^2}\right|_{x_0} \left[\frac{\Delta x}{2}\right]^2 + O^-(\Delta x^3/8) \tag{7.12}$$

Subtracting Equation 7.12 from Equation 7.11, and then solving for the differential $\partial f/\partial x$, produces Equation 7.13,

$$\left.\frac{\partial f}{\partial x}\right|_{x_0} = \frac{f(x_0 + \frac{\Delta x}{2}) - f(x_0 - \frac{\Delta x}{2})}{\Delta x} + O(\Delta x^2) \tag{7.13}$$

where $O(\Delta x^2) = (O^-(\Delta x^3/8) - O^+(\Delta x^3/8))/\Delta x \propto \Delta x^2$. The above scheme is called **central difference in space** [2, 77]. From Equation 7.9, the 1^{st} degree Taylor polynomial of the function $f(t)$ at the time point $t_0 + \Delta t/2$ is given by Equation 7.14.

$$f(t_0 + \frac{\Delta t}{2}) = f(t_0) + \left.\frac{\partial f}{\partial t}\right|_{t_0} \frac{\Delta t}{2} + O^+(\Delta t^2/4) \tag{7.14}$$

Solving for the differential $\partial f/\partial t$ in the equation above, produces the following equation:

$$\left.\frac{\partial f}{\partial t}\right|_{t_0} = \frac{f(t_0 + \frac{\Delta t}{2}) - f(t_0)}{\Delta t/2} + O(\Delta t) \tag{7.15}$$

where $O(\Delta t) = -O^+(\Delta t^2/4)/(\Delta t/2) \propto \Delta t$. The above scheme is called **forward difference in time** [2, 77].

7.3.2 Discretization of $T_{r\theta}$ and $T_{z\theta}$

When discretizing the terms $T_{r\theta}$ and $T_{z\theta}$, central difference in space is used. According to Equation 7.13, this scheme will produce a truncation error of $O(\Delta r^2) + O(\Delta z^2)$. Because of this type of error (see Section 7.6) plus any errors introduced by the numerical treatment of the boundary conditions, it is to be expected that the (unknown)

correct solution $v_\theta(r,z,t)$ of the partial differential Equation 7.6 will differ from the **exact solution**[1] $v^{\mathrm{k}}_{\mathrm{i,j}}$ of the system of algebraic Equations 7.28 to 7.31, shown later [2, 33]. This difference is represented with the term $e^{\mathrm{k}}_{\mathrm{ij}} = v_\theta(r,z,t) - v^{\mathrm{k}}_{\mathrm{i,j}}$ and is known as the **discretization error** [2]. The discrete counterpart of the terms $v_\theta(r,z,t)$, $\dot\gamma(r,z,t)$, $\eta(r,z,t)$, $T_{r\theta}(r,z,t)$ and $T_{z\theta}(r,z,t)$ are represented with $v^{\mathrm{k}}_{\mathrm{i,j}}$, $\dot\gamma^{\mathrm{k}}_{\mathrm{i,j}}$, $\eta^{\mathrm{k}}_{\mathrm{i,j}}$, $T^{\mathrm{k}}_{\mathrm{i,j}}$ and $Z^{\mathrm{k}}_{\mathrm{i,j}}$, respectively. Now, discretizing Equation 7.7 according to the method of finite differences, with central difference in space, gives Equation 7.16:

$$\begin{aligned} T_{\mathrm{i,j}} &= \eta_{\mathrm{i,j}}\left[\frac{v_{\mathrm{i}+\frac{1}{2},\mathrm{j}} - v_{\mathrm{i}-\frac{1}{2},\mathrm{j}}}{\Delta r} - \frac{v_{\mathrm{i,j}}}{r_{\mathrm{i}}}\right] = \eta_{\mathrm{i,j}}\left[\frac{v_{\mathrm{i}+\frac{1}{2},\mathrm{j}} - v_{\mathrm{i}-\frac{1}{2},\mathrm{j}}}{\Delta r} - \frac{(v_{\mathrm{i}+\frac{1}{2},\mathrm{j}} + v_{\mathrm{i}-\frac{1}{2},\mathrm{j}})/2}{(r_{\mathrm{i}+\frac{1}{2}} + r_{\mathrm{i}-\frac{1}{2}})/2}\right] \\ &= \eta_{\mathrm{i,j}}\left[\frac{1}{\Delta r} - \frac{1}{r_{\mathrm{i}+\frac{1}{2}} + r_{\mathrm{i}-\frac{1}{2}}}\right] v_{\mathrm{i}+\frac{1}{2},\mathrm{j}} - \eta_{\mathrm{i,j}}\left[\frac{1}{\Delta r} + \frac{1}{r_{\mathrm{i}+\frac{1}{2}} + r_{\mathrm{i}-\frac{1}{2}}}\right] v_{\mathrm{i}-\frac{1}{2},\mathrm{j}} \end{aligned} \tag{7.16}$$

Since the origin $(r,z,t) = (0,0,0)$ starts at the grid point $(\mathrm{i},\mathrm{j},\mathrm{k}) = (1,1,0)$, then $r_{\mathrm{i}} = (\mathrm{i}-1)\,\Delta r$, $z_{\mathrm{j}} = (\mathrm{j}-1)\,\Delta z$ and $t_{\mathrm{k}} = \mathrm{k}\,\Delta t$. Using Equation 7.16 at the grid points (i+$\frac{1}{2}$,j) and (i-$\frac{1}{2}$,j) produces Equations 7.17 and 7.18.

$$\begin{aligned} T_{\mathrm{i}+\frac{1}{2},\mathrm{j}} &= \eta_{\mathrm{i}+\frac{1}{2},\mathrm{j}}\left[\frac{1}{\Delta r} - \frac{1}{r_{\mathrm{i}+1} + r_{\mathrm{i}}}\right] v_{\mathrm{i}+1,\mathrm{j}} - \eta_{\mathrm{i}+\frac{1}{2},\mathrm{j}}\left[\frac{1}{\Delta r} + \frac{1}{r_{\mathrm{i}+1} + r_{\mathrm{i}}}\right] v_{\mathrm{i,j}} \\ &= \Xi_{\mathrm{i}+1,\mathrm{j}} v_{\mathrm{i}+1,\mathrm{j}} - \Theta_{\mathrm{i}+1,\mathrm{j}} v_{\mathrm{i,j}} \end{aligned} \tag{7.17}$$

$$\begin{aligned} T_{\mathrm{i}-\frac{1}{2},\mathrm{j}} &= \eta_{\mathrm{i}-\frac{1}{2},\mathrm{j}}\left[\frac{1}{\Delta r} - \frac{1}{r_{\mathrm{i}} + r_{\mathrm{i}-1}}\right] v_{\mathrm{i,j}} - \eta_{\mathrm{i}-\frac{1}{2},\mathrm{j}}\left[\frac{1}{\Delta r} + \frac{1}{r_{\mathrm{i}} + r_{\mathrm{i}-1}}\right] v_{\mathrm{i}-1,\mathrm{j}} \\ &= \Xi_{\mathrm{i,j}} v_{\mathrm{i,j}} - \Theta_{\mathrm{i,j}} v_{\mathrm{i}-1,\mathrm{j}} \end{aligned} \tag{7.18}$$

Figure 7.2 demonstrates a schematic figure of the grid system used in the above (and all the remaining) discretization.

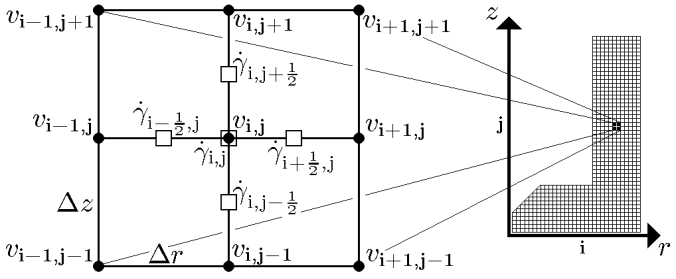


Figure 7.2: Schematic figure of the grid system used in this work (see also Figures 8.1, 8.2 and 10.22 on Pages 179, 180 and 256, respectively).

The discrete presentation of Equation 7.7, at the grid point (i,j) is shown by the

[1]The **exact solution** of the system of algebraic equations is obtained when no round-off error are present in the numerical calculation. Such error exists because of the inaccuracy introduced after a repetitive number of calculations in which the computer is constantly rounding the numbers to some significant figure [2, 33].

following equation:

$$\begin{aligned} T_{\mathrm{i,j}} &= \eta_{\mathrm{i,j}} \left[\frac{v_{\mathrm{i}+\frac{1}{2},\mathrm{j}} - v_{\mathrm{i}-\frac{1}{2},\mathrm{j}}}{\Delta r} - \frac{v_{\mathrm{i,j}}}{r_{\mathrm{i}}} \right] = \eta_{\mathrm{i,j}} \left[\frac{\frac{1}{2}(v_{\mathrm{i}+1,\mathrm{j}} + v_{\mathrm{i,j}}) - \frac{1}{2}(v_{\mathrm{i,j}} + v_{\mathrm{i}-1,\mathrm{j}})}{\Delta r} - \frac{v_{\mathrm{i,j}}}{r_{\mathrm{i}}} \right] \\ &= \eta_{\mathrm{i,j}} \left[\frac{v_{\mathrm{i}+1,\mathrm{j}} - v_{\mathrm{i}-1,\mathrm{j}}}{2\,\Delta r} - \frac{v_{\mathrm{i,j}}}{r_{\mathrm{i}}} \right] = \frac{\eta_{\mathrm{i,j}}}{2\,\Delta r} v_{\mathrm{i}+1,\mathrm{j}} - \frac{\eta_{\mathrm{i,j}}}{r_{\mathrm{i}}} v_{\mathrm{i,j}} - \frac{\eta_{\mathrm{i,j}}}{2\,\Delta r} v_{\mathrm{i}-1,\mathrm{j}} \\ &= \Psi_{\mathrm{i,j}} v_{\mathrm{i}+1,\mathrm{j}} - \Lambda_{\mathrm{i,j}} v_{\mathrm{i,j}} - \Psi_{\mathrm{i,j}} v_{\mathrm{i}-1,\mathrm{j}} \end{aligned} \tag{7.19}$$

Also, for the term Z at the grid point (i,j+$\frac{1}{2}$) and (i,j-$\frac{1}{2}$), a central difference is applied as demonstrated with Equations 7.20 and 7.21.

$$\begin{aligned} Z_{\mathrm{i,j}+\frac{1}{2}} &= \eta_{\mathrm{i,j}+\frac{1}{2}} \frac{v_{\mathrm{i,j}+1} - v_{\mathrm{i,j}}}{\Delta z} = \frac{\eta_{\mathrm{i,j}+\frac{1}{2}}}{\Delta z} v_{\mathrm{i,j}+1} - \frac{\eta_{\mathrm{i,j}+\frac{1}{2}}}{\Delta z} v_{\mathrm{i,j}} \\ &= \Upsilon_{\mathrm{i,j}+1} v_{\mathrm{i,j}+1} - \Upsilon_{\mathrm{i,j}+1} v_{\mathrm{i,j}} \end{aligned} \tag{7.20}$$

$$\begin{aligned} Z_{\mathrm{i,j}-\frac{1}{2}} &= \eta_{\mathrm{i,j}-\frac{1}{2}} \frac{v_{\mathrm{i,j}} - v_{\mathrm{i,j}-1}}{\Delta z} = \frac{\eta_{\mathrm{i,j}-\frac{1}{2}}}{\Delta z} v_{\mathrm{i,j}} - \frac{\eta_{\mathrm{i,j}-\frac{1}{2}}}{\Delta z} v_{\mathrm{i,j}-1} \\ &= \Upsilon_{\mathrm{i,j}} v_{\mathrm{i,j}} - \Upsilon_{\mathrm{i,j}} v_{\mathrm{i,j}-1} \end{aligned} \tag{7.21}$$

7.4 Numerical Governing Equation

As applies for the terms T and Z, a central difference in space is used when discretizing the governing Equation 7.6. For the time stepping, the **Alternating Direction Implicit (ADI)** technique [2, 33, 77] is used, mainly to reduce memory consumption, and also to be able to utilize the well known **Thomas algorithm** [2, 33]. When solving a system of N algebraic equations, the Thomas algorithm is quite economical and requires only $5\,\mathrm{N} - 4$ operations, i.e. multiplications and divisions [33]. The time step is a forward differencing scheme (see Equation 7.15) with an accuracy of $O(\Delta t)$.

With the ADI technique, the governing Equation 7.6 is now presented with two different algebraic equations:

$$\rho \frac{v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - v_{\mathrm{i,j}}^{\mathrm{k}}}{\Delta t/2} = \frac{T_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} - T_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}}}{\Delta r} + 2\,\frac{T_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}}{r_{\mathrm{i}}} + \frac{Z_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}} - Z_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}}}{\Delta z} \tag{7.22}$$

$$\rho \frac{v_{\mathrm{i,j}}^{\mathrm{k}+1} - v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}}{\Delta t/2} = \frac{T_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} - T_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}}}{\Delta r} + 2\,\frac{T_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}}{r_{\mathrm{i}}} + \frac{Z_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}+1} - Z_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}+1}}{\Delta z} \tag{7.23}$$

By defining a new variable and two constants, namely $\theta_{\mathrm{i}} = \Delta t/r_{\mathrm{i}}$, $\beta = \Delta t/(2\,\Delta r)$ and $\chi = \Delta t/(2\,\Delta z)$, and using them when rearranging the two above equations, Equations 7.24 and 7.25 are produced.

$$v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - v_{\mathrm{i,j}}^{\mathrm{k}} = \frac{\beta}{\rho} \left[T_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} - T_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} \right] + \frac{\theta_{\mathrm{i}}}{\rho} T_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\chi}{\rho} \left[Z_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}} - Z_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}} \right] \tag{7.24}$$

$$v_{\mathrm{i,j}}^{\mathrm{k}+1} - v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = \frac{\beta}{\rho} \left[T_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} - T_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} \right] + \frac{\theta_{\mathrm{i}}}{\rho} T_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\chi}{\rho} \left[Z_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}+1} - Z_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}+1} \right] \tag{7.25}$$

Substituting the result from Equations 7.17 to 7.21 into the above and then rearranging in order to isolate the individual velocity terms, gives the following two equations:

$$\begin{aligned} v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - v_{\mathrm{i,j}}^{\mathrm{k}} = & \left[\frac{\beta}{\rho}\Xi_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\theta_{\mathrm{i}}}{\rho}\Psi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}\right] v_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \\ & - \left[\frac{\beta}{\rho}\Theta_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\beta}{\rho}\Xi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\theta_{\mathrm{i}}}{\rho}\Lambda_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}\right] v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \\ & + \left[\frac{\beta}{\rho}\Theta_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - \frac{\theta_{\mathrm{i}}}{\rho}\Psi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}\right] v_{\mathrm{i}-1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \left[\frac{\chi}{\rho}\Upsilon_{\mathrm{i,j}+1}^{\mathrm{k}}\right] v_{\mathrm{i,j}+1}^{\mathrm{k}} + \\ & - \left[\frac{\chi}{\rho}(\Upsilon_{\mathrm{i,j}+1}^{\mathrm{k}} + \Upsilon_{\mathrm{i,j}}^{\mathrm{k}})\right] v_{\mathrm{i,j}}^{\mathrm{k}} + \left[\frac{\chi}{\rho}\Upsilon_{\mathrm{i,j}}^{\mathrm{k}}\right] v_{\mathrm{i,j}-1}^{\mathrm{k}} \end{aligned} \tag{7.26}$$

$$\begin{aligned} v_{\mathrm{i,j}}^{\mathrm{k}+1} - v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = & \left[\frac{\beta}{\rho}\Xi_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\theta_{\mathrm{i}}}{\rho}\Psi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}\right] v_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \\ & - \left[\frac{\beta}{\rho}\Theta_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\beta}{\rho}\Xi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\theta_{\mathrm{i}}}{\rho}\Lambda_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}\right] v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \\ & + \left[\frac{\beta}{\rho}\Theta_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - \frac{\theta_{\mathrm{i}}}{\rho}\Psi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}\right] v_{\mathrm{i}-1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \left[\frac{\chi}{\rho}\Upsilon_{\mathrm{i,j}+1}^{\mathrm{k}+1}\right] v_{\mathrm{i,j}+1}^{\mathrm{k}+1} + \\ & - \left[\frac{\chi}{\rho}(\Upsilon_{\mathrm{i,j}+1}^{\mathrm{k}+1} + \Upsilon_{\mathrm{i,j}}^{\mathrm{k}+1})\right] v_{\mathrm{i,j}}^{\mathrm{k}+1} + \left[\frac{\chi}{\rho}\Upsilon_{\mathrm{i,j}}^{\mathrm{k}+1}\right] v_{\mathrm{i,j}-1}^{\mathrm{k}+1} \end{aligned} \tag{7.27}$$

To simplify the above nomenclature, new quantities A, B, C, D, E, F, K and L are defined as demonstrated with Equations from 7.32 to 7.37. With those new definitions, Equation 7.26 can now be rewritten to the following:

$$A_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} - (1 + B_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}) v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + C_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i}-1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} = K_{\mathrm{i,j}}^{\mathrm{k}} \tag{7.28}$$

where the term $K_{\mathrm{i,j}}^{\mathrm{k}}$ is given by Equation 7.29.

$$K_{\mathrm{i,j}}^{\mathrm{k}} = -D_{\mathrm{i,j}}^{\mathrm{k}} v_{\mathrm{i,j}+1}^{\mathrm{k}} - (1 - E_{\mathrm{i,j}}^{\mathrm{k}}) v_{\mathrm{i,j}}^{\mathrm{k}} - F_{\mathrm{i,j}}^{\mathrm{k}} v_{\mathrm{i,j}-1}^{\mathrm{k}} \tag{7.29}$$

In the same manner, Equation 7.27 can be rewritten as:

$$D_{\mathrm{i,j}}^{\mathrm{k}+1} v_{\mathrm{i,j}+1}^{\mathrm{k}+1} - (1 + E_{\mathrm{i,j}}^{\mathrm{k}+1}) v_{\mathrm{i,j}}^{\mathrm{k}+1} + F_{\mathrm{i,j}}^{\mathrm{k}+1} v_{\mathrm{i,j}-1}^{\mathrm{k}+1} = L_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \tag{7.30}$$

where the term $L_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}$ is given by Equation 7.31.

$$L_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = -A_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} - (1 - B_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}) v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - C_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i}-1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} \tag{7.31}$$

The important quantities A, B, C, D, E and F are as follows:

$$A_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = \frac{\beta}{\rho}\Xi_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\theta_{\mathrm{i}}}{\rho}\Psi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = \tilde{\beta}\, \eta_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} \left[\frac{1}{\Delta r} - \frac{1}{r_{\mathrm{i}+1} + r_{\mathrm{i}}}\right] + \tilde{\theta}_{\mathrm{i}}\, \frac{\eta_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}}{2\,\Delta r} \tag{7.32}$$

$$\begin{aligned} B_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = & \frac{\beta}{\rho}\Theta_{\mathrm{i}+1,\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\beta}{\rho}\Xi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + \frac{\theta_{\mathrm{i}}}{\rho}\Lambda_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = \tilde{\beta}\, \eta_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} \left[\frac{1}{\Delta r} + \frac{1}{r_{\mathrm{i}+1} + r_{\mathrm{i}}}\right] + \\ & + \tilde{\beta}\, \eta_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}} \left[\frac{1}{\Delta r} - \frac{1}{r_{\mathrm{i}} + r_{\mathrm{i}-1}}\right] + \tilde{\theta}_{\mathrm{i}}\, \frac{\eta_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}}{r_{\mathrm{i}}} \end{aligned} \tag{7.33}$$

$$C_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = \frac{\beta}{\rho}\Theta_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - \frac{\theta_{\mathrm{i}}}{\rho}\Psi_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = \tilde{\beta}\,\eta_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^{\mathrm{k}+\frac{1}{2}}\left[\frac{1}{\Delta r} + \frac{1}{r_{\mathrm{i}} + r_{\mathrm{i}-1}}\right] - \tilde{\theta}_{\mathrm{i}}\,\frac{\eta_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}}{2\,\Delta r} \tag{7.34}$$

$$D_{\mathrm{i,j}}^{\mathrm{k}+1} = \frac{\chi}{\rho}\Upsilon_{\mathrm{i,j}+1}^{\mathrm{k}+1} = \tilde{\chi}\,\frac{\eta_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}+1}}{\Delta z} \quad \wedge \quad D_{\mathrm{i,j}}^{\mathrm{k}} = \tilde{\chi}\,\frac{\eta_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}}}{\Delta z} \tag{7.35}$$

$$E_{\mathrm{i,j}}^{\mathrm{k}+1} = \frac{\chi}{\rho}(\Upsilon_{\mathrm{i,j}+1}^{\mathrm{k}+1} + \Upsilon_{\mathrm{i,j}}^{\mathrm{k}+1}) = \tilde{\chi}\,\frac{\eta_{\mathrm{i,j}+\frac{1}{2}}^{\mathrm{k}+1}}{\Delta z} + \tilde{\chi}\,\frac{\eta_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}+1}}{\Delta z} \quad \wedge \quad E_{\mathrm{i,j}}^{\mathrm{k}} = \cdots \tag{7.36}$$

$$F_{\mathrm{i,j}}^{\mathrm{k}+1} = \frac{\chi}{\rho}\Upsilon_{\mathrm{i,j}}^{\mathrm{k}+1} = \tilde{\chi}\,\frac{\eta_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}+1}}{\Delta z} \quad \wedge \quad F_{\mathrm{i,j}}^{\mathrm{k}} = \tilde{\chi}\,\frac{\eta_{\mathrm{i,j}-\frac{1}{2}}^{\mathrm{k}}}{\Delta z} \tag{7.37}$$

In the above, additional new quantities have been defined: $\tilde{\beta} = \beta/\rho = \Delta t/(2\,\Delta r\,\rho)$, $\tilde{\chi} = \chi/\rho = \Delta t/(2\,\Delta z\,\rho)$ and $\tilde{\theta}_{\mathrm{i}} = \Delta t/(r_{\mathrm{i}}\,\rho)$.

7.5 Shear Rate

From Equations 7.32 to 7.37 it is apparent that information about the shear viscosity η at the grid points (i+$\frac{1}{2}$,j) and (i-$\frac{1}{2}$,j) (i,j) (i,j+$\frac{1}{2}$) and (i,j-$\frac{1}{2}$) is needed. Since $\eta_{\mathrm{i,j}} = \eta(\dot{\gamma}_{\mathrm{i,j}}, \tilde{\Gamma}_{\mathrm{i,j}}, \tilde{\Theta}_{\mathrm{i,j}}, k\,\Delta t)$, one must first calculate the shear rate $\dot{\gamma}_{\mathrm{i,j}}$ at the corresponding points (the terms $\tilde{\Gamma}_{\mathrm{i,j}}$ and $\tilde{\Theta}_{\mathrm{i,j}}$ are memory modules, defined in Equations 7.63 and 7.64). Going through a similar procedure as was done for Equations 7.16 to 7.21, when discretizing Equation 7.3, produces the following (see also Figure 7.2):

$$\dot{\gamma}_{\mathrm{i,j}} = \sqrt{\left[\frac{v_{\mathrm{i}+1,\mathrm{j}} - v_{\mathrm{i}-1,\mathrm{j}}}{2\,\Delta r} - \frac{v_{\mathrm{i,j}}}{r_{\mathrm{i}}}\right]^2 + \left[\frac{v_{\mathrm{i,j}+1} - v_{\mathrm{i,j}-1}}{2\,\Delta z}\right]^2} \tag{7.38}$$

$$\begin{aligned}\dot{\gamma}_{\mathrm{i}+\frac{1}{2},\mathrm{j}}^2 = {} & \left[\frac{v_{\mathrm{i}+1,\mathrm{j}} - v_{\mathrm{i,j}}}{\Delta r} - \frac{v_{\mathrm{i}+1,\mathrm{j}} + v_{\mathrm{i,j}}}{r_{\mathrm{i}+1} + r_{\mathrm{i}}}\right]^2 + \\ & + \left[\frac{(v_{\mathrm{i}+1,\mathrm{j}+1} + v_{\mathrm{i,j}+1})/2 - (v_{\mathrm{i}+1,\mathrm{j}-1} + v_{\mathrm{i,j}-1})/2}{2\,\Delta z}\right]^2\end{aligned} \tag{7.39}$$

$$\begin{aligned}\dot{\gamma}_{\mathrm{i}-\frac{1}{2},\mathrm{j}}^2 = {} & \left[\frac{v_{\mathrm{i,j}} - v_{\mathrm{i}-1,\mathrm{j}}}{\Delta r} - \frac{v_{\mathrm{i,j}} + v_{\mathrm{i}-1,\mathrm{j}}}{r_{\mathrm{i}} + r_{\mathrm{i}-1}}\right]^2 + \\ & + \left[\frac{(v_{\mathrm{i,j}+1} + v_{\mathrm{i}-1,\mathrm{j}+1})/2 - (v_{\mathrm{i,j}-1} + v_{\mathrm{i}-1,\mathrm{j}-1})/2}{2\,\Delta z}\right]^2\end{aligned} \tag{7.40}$$

$$\begin{aligned}\dot{\gamma}_{\mathrm{i,j}+\frac{1}{2}}^2 = {} & \left[\frac{(v_{\mathrm{i}+1,\mathrm{j}+1} + v_{\mathrm{i}+1,\mathrm{j}})/2 - (v_{\mathrm{i}-1,\mathrm{j}+1} + v_{\mathrm{i}-1,\mathrm{j}})/2}{2\,\Delta r} - \frac{v_{\mathrm{i,j}+1} + v_{\mathrm{i,j}}}{2\,r_{\mathrm{i}}}\right]^2 + \\ & + \left[\frac{v_{\mathrm{i,j}+1} - v_{\mathrm{i,j}}}{\Delta z}\right]^2\end{aligned} \tag{7.41}$$

$$\begin{aligned}\dot{\gamma}_{\mathrm{i,j}-\frac{1}{2}}^2 = {} & \left[\frac{(v_{\mathrm{i}+1,\mathrm{j}} + v_{\mathrm{i}+1,\mathrm{j}-1})/2 - (v_{\mathrm{i}-1,\mathrm{j}} + v_{\mathrm{i}-1,\mathrm{j}-1})/2}{2\,\Delta r} - \frac{v_{\mathrm{i,j}} + v_{\mathrm{i,j}-1}}{2\,r_{\mathrm{i}}}\right]^2 + \\ & + \left[\frac{v_{\mathrm{i,j}} - v_{\mathrm{i,j}-1}}{\Delta z}\right]^2\end{aligned} \tag{7.42}$$

7.6 Consistency

When all the partial derivatives in Equations 7.6, 7.7 and 7.8 are replaced by the finite differences quotients of Equations 7.13 and 7.15, then with the **ADI** technique, the result is two algebraic equations, namely Equations 7.22 and 7.23. These two equations are equivalent to the set of algebraic Equations 7.28 to 7.31 and as such, the following discussion applies equally to the last two mentioned equations. All these algebraic equations are called **difference equations** [2]. They are an algebraic representation of the PDE to be solved and is just an approximation of it, due to the truncation error produced in the transformation. In other words, the difference equations are not the same as the original PDE as shown with the following: **Original PDE** = **System of Algebraic Equations** + **Truncation Error**.

Substituting the terms of Equations 7.7 and 7.8 into Equation 7.6, without expanding the velocity terms, produces the **Original PDE**, shown with Equation 7.58. Making the same discretization on it as was done to produce either Equation 7.22 or 7.23, results in Equation 7.43. It should be noted that Equations 7.22 and 7.23 are identical to the algebraic part of Equation 7.43.

$$
\begin{aligned}
0 &= \rho\frac{\partial v_\theta}{\partial t} - \frac{\partial}{\partial r}\left[\eta(v_\theta)\left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right)\right] - \frac{2\,\eta(v_\theta)}{r}\left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right) - \frac{\partial}{\partial z}\left(\eta(v_\theta)\frac{\partial v_\theta}{\partial z}\right) \\
&= \rho\frac{v(t_\mathrm{o}+\frac{\Delta t}{2}) - v(t_\mathrm{o})}{\Delta t/2} + O(\Delta t) + \\
&- \frac{\left[\eta\left(\frac{v(r_\mathrm{o}+\Delta r)-v(r_\mathrm{o})}{\Delta r} - \frac{v}{r}\right)\right]\Big|_{r_\mathrm{o}+\frac{\Delta r}{2}} - \left[\eta\left(\frac{v(r_\mathrm{o})-v(r_\mathrm{o}-\Delta r)}{\Delta r} - \frac{v}{r}\right)\right]\Big|_{r_\mathrm{o}-\frac{\Delta r}{2}} + O(\Delta r^2)}{\Delta r} + \\
&- O(\Delta r^2) - \frac{2\,\eta(r_\mathrm{o})}{r_\mathrm{o}}\left[\frac{v(r_\mathrm{o}+\frac{\Delta r}{2}) - v(r_\mathrm{o}-\frac{\Delta r}{2})}{\Delta r} - \frac{v(r_\mathrm{o})}{r_\mathrm{o}}\right] - O(\Delta r^2) + \\
&- \frac{\left[\eta\left(\frac{v(z_\mathrm{o}+\Delta z)-v(z_\mathrm{o})}{\Delta z}\right)\right]\Big|_{z_\mathrm{o}+\frac{\Delta z}{2}} - \left[\eta\left(\frac{v(z_\mathrm{o})-v(z_\mathrm{o}-\Delta z)}{\Delta z}\right)\right]\Big|_{z_\mathrm{o}-\frac{\Delta z}{2}} + O(\Delta z^2)}{\Delta z} - O(\Delta z^2)
\end{aligned}
\tag{7.43}
$$

In the above equation, the largest truncation error controls the actual truncation error. As shown, it consists of $O(\Delta r^2)/\Delta r$, $O(\Delta z^2)/\Delta z$ and $O(\Delta t)$, i.e. the difference between the **Original PDE** and the **System of Algebraic Equations** is $O(\Delta t) + O(\Delta r) + O(\Delta z) \propto \Delta t + \Delta r + \Delta z$. It is apparent from this, that as the number of grid points goes to infinity ($\Delta t \to 0, \Delta r \to 0$ and $\Delta z \to 0$), both algebraic Equations 7.22 and 7.23 (or equally, Equations 7.28 and 7.30) become equivalent to the original partial differential Equation 7.6. With this characteristic, the finite difference representation of the PDE, namely Equations 7.22 and 7.23, is said to be **consistent** [2, 33, 77].

In this work, the grid contains uniform spacing, Δr = constant and Δz = constant. The majority of computational fluid dynamics (CFD) applications involve numerical solutions on a grid which contains uniform spacing[2] in each directions because this greatly simplifies the programming of the solution, saves storage space, and usually results in greater accuracy [2]. When calculating the fluid flow inside the ConTec Viscometer 4, the spacing between grid points is set to be $\Delta r = \Delta z = 0.5\,\mathrm{mm}$. When solving for the ConTec **BML** Viscometer 3 and for the BTRHEOM, the

[2]This uniform spacing does not have to occur in the physical space. Often the numerical calculations are carried out in a transformed computational space that has uniform spacing and corresponds to a nonuniform spacing in the physical plane [2].

values of $\Delta r = \Delta z = 1\,\text{mm}$ are used. For the prototype viscometers, two different conditions are applied, namely $\Delta r = \Delta z = 2\,\text{mm}$ and $\Delta r = 2\,\text{mm} \wedge \Delta z = 1\,\text{mm}$.

7.7 Auxiliary (Boundary and Initial) Conditions

As shown in Figure 7.3, the solution region which Equation 7.6 is solved, is designated Ω. The boundary that envelopes this area is designated $\partial\Omega$. Hence the total solution region is $\Omega_{\text{tot}} \in \Omega \cup \partial\Omega$. There are three types of boundary conditions used in this work, namely the Dirichlet-, Neumann- and Robin boundary condition. Their corresponding location is designated with $\partial\Omega_{\text{D}}$, $\partial\Omega_{\text{N}}$ and $\partial\Omega_{\text{R}}$, respectively.

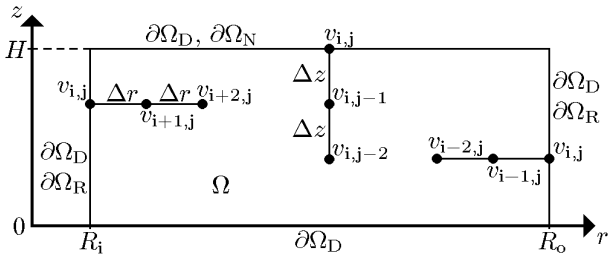


Figure 7.3: Schematic figure of the grid system and the corresponding boundary conditions to be used: $\partial\Omega_{\text{D}}$, $\partial\Omega_{\text{N}}$ and $\partial\Omega_{\text{R}}$.

Dirichlet Boundary Condition $\partial\Omega_{\text{D}}$ (ConTec, BTRHEOM & prototype)

The Dirichlet boundary condition is the easiest condition to implement since it consist of setting pre-known values v_0 at the edge of the solution region: $v_\theta(r,z,t) = v^{\text{k}}_{\text{i,j}} = v_0(r,t)\ \forall\ (r,z) \in \partial\Omega_{\text{D}}$. According to Fletcher [33], for the ADI transient problem (i.e. ADI with time dependence), the direct implementation of the boundary condition, as shown here, will lead to an error of $O(\Delta t)$. If it is desired to gain an accuracy of $O(\Delta t^2)$, another type of implementation is required. Since the time step already has the accuracy of only $O(\Delta t)$ (c.f. Equation 7.43), such treatment is not required here.

The Dirichlet boundary condition is the one that is most used. For the time independent problem, it consists of $v_\theta(r,z) = v_{\text{i,j}} = v_0(r)$, and for the transient problem, the condition consist of $v_\theta(r,z,t) = v^{\text{k}}_{\text{i,j}} = v_0(r,t)\ \forall\ (r,z) \in \partial\Omega_{\text{D}}$.

Neumann Boundary Condition $\partial\Omega_{\text{N}}$ (ConTec & prototype)

In this work, the Neumann boundary condition consists of $\partial v_\theta(r,z,t)/\partial z = 0\ \forall\ (r,z) \in \partial\Omega_{\text{N}}$. Applying the central difference in space (see Equation 7.13) when converting the above boundary condition into its algebraic counter part, gives $(v_{\text{i,j+1}} - v_{\text{i,j-1}})/(2\,\Delta z) = 0 \Rightarrow v_{\text{i,j+1}} = v_{\text{i,j-1}}$. With this result in mind and comparing Equation 7.41 with Equation 7.42, it is evident that $\dot\gamma_{\text{i,j+}\frac{1}{2}} = \dot\gamma_{\text{i,j-}\frac{1}{2}}$, which furthermore leads to $\eta_{\text{i,j+}\frac{1}{2}} = \eta_{\text{i,j-}\frac{1}{2}}$. According to Equations 7.35 and 7.37, this gives $D_{\text{i,j}} = F_{\text{i,j}}$.

Equations 7.28 and 7.31 remain unchanged when implementing the above boundary condition. But in the case of Equations 7.29 and 7.30, the term $D_{\text{i,j}} v_{\text{i,j+1}}$ is

replaced with $F_{\mathrm{i,j}} v_{\mathrm{i,j-1}}$. Hence, the set of algebraic equations to be solved where Neumann boundary condition rules, is given by Equations 7.44 to 7.47.

$$A_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i+1,j}}^{\mathrm{k}+\frac{1}{2}} - (1 + B_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}) v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} + C_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i-1,j}}^{\mathrm{k}+\frac{1}{2}} = K_{\mathrm{i,j}}^{\mathrm{k}} \tag{7.44}$$

$$K_{\mathrm{i,j}}^{\mathrm{k}} = -(1 - E_{\mathrm{i,j}}^{\mathrm{k}}) v_{\mathrm{i,j}}^{\mathrm{k}} - 2\, F_{\mathrm{i,j}}^{\mathrm{k}} v_{\mathrm{i,j-1}}^{\mathrm{k}} \quad \wedge \quad v_{\mathrm{i,j+1}}^{\mathrm{k}} = v_{\mathrm{i,j-1}}^{\mathrm{k}} \tag{7.45}$$

$$-(1 + E_{\mathrm{i,j}}^{\mathrm{k+1}}) v_{\mathrm{i,j}}^{\mathrm{k+1}} + 2\, F_{\mathrm{i,j}}^{\mathrm{k+1}} v_{\mathrm{i,j-1}}^{\mathrm{k+1}} = L_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \quad \wedge \quad v_{\mathrm{i,j+1}}^{\mathrm{k+1}} = v_{\mathrm{i,j-1}}^{\mathrm{k+1}} \tag{7.46}$$

$$L_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} = -A_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i+1,j}}^{\mathrm{k}+\frac{1}{2}} - (1 - B_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}) v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} - C_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} v_{\mathrm{i-1,j}}^{\mathrm{k}+\frac{1}{2}} \tag{7.47}$$

Robin Boundary Condition $\partial\Omega_{\mathrm{R}}$ (BTRHEOM; "Full-slippage-case")

Here, the Robin condition consists of $\partial v_\theta(r,z,t)/\partial r = v_\theta/r \;\forall\; (r,z) \in \partial\Omega_{\mathrm{R}}$. Rather than implementing this condition algebraically in similar way as was done in Equations 7.44 to 7.47, it is enforced numerically. Whenever the Robin condition occurs, the corresponding boundary values are first calculated according to Equations 7.56 and 7.57 and thereafter treated as Dirichlet condition when solving $\mathbf{A}\cdot\mathbf{v}^{\mathrm{k+1}} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$. This type of implementation is unconventional, and it nourishes very much on the successive substitution method, described in Section 7.8. The nourishment consists of that the boundary condition is recalculated (iterated) and hence (hopefully) refined towards the correct value, for each *successive* iteration step. The calculation for the next time step k+2 is not activated until the boundary value has reached some kind of equilibrium value. Hence, both the solution of Equation 7.6 and the boundary condition have to be stable if the next time step is to be calculated.

In the early stages of code development for the ConTec viscometers, the same approach was used for the Neumann condition ($\partial\Omega_{\mathrm{N}}$). This approach was implemented with Equation 7.55 and produced a small error at and near the corresponding boundary. Rather to have the correct condition $\partial v_\theta/\partial r = 0$ at this zone, the peculiar condition $\partial v_\theta/\partial r \leq 0$ was produced. This is shown with Figure 7.4, which demonstrates an isoplot of the relative difference between velocity in the ConTec Viscometer 4 after ($\mathbf{v}_{\mathrm{new}}$) and before ($\mathbf{v}_{\mathrm{old}}$) implementation of Equations 7.44 to 7.47. The isolines consist of $((\mathbf{v}_{\mathrm{new}} - \mathbf{v}_{\mathrm{old}})/[\frac{1}{2}(\mathbf{v}_{\mathrm{new}} + \mathbf{v}_{\mathrm{old}})])$-values. The number of isolines are 12 and have the following values: $[-2, -0.5, 0.1, 0.25, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4]\cdot 10^{-2}$, i.e. the difference is in the order of magnitude of 1%. The flowing material used in this analysis, consist of a Newtonian fluid, with the shear viscosity of $\eta = 3\,\mathrm{Pa}\cdot\mathrm{s}$. The rotational frequency applied for this example is $f_{\mathrm{o}} = 0.3\,\mathrm{rps}$.

The only time the Robin boundary condition is used, is when verifying a known solution. Therefore, this crude and simple implementation can be accepted.

Implementation of the Robin Boundary Condition

As shown in the schematic Figure 7.3, at the boundary $\partial\Omega$ there is only one direction to go, namely the direction away from it, into the solution area Ω. To maintain the same accuracy as is achieved with Equation 7.13, a polynomial approach is used to create the algebraic counterpart of the partial derivative $\partial v_\theta/\partial r$. Assuming that the velocity v_θ, at the random location r, can be expressed with the polynomial

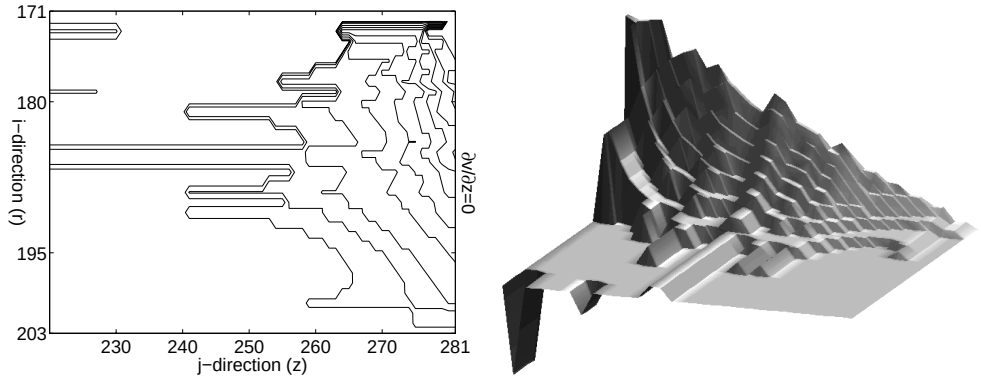


Figure 7.4: Isoplot of the relative difference between velocity in the ConTec Viscometer 4 (see Figure 8.2 on Page 180 for orientation) after ($\mathbf{v}_{\mathrm{new}}$) and before ($\mathbf{v}_{\mathrm{old}}$) implementation of Equations 7.44 to 7.47. To the right is a three dimensional representation of the isoplot.

$v_\theta = a + b\,r + c\,r^2$, its corresponding velocity $v_{\mathrm{i,j}}$ at the boundary $r = R_{\mathrm{i}}$, can be represented with Equation 7.48.

$$v_{\mathrm{i,j}} = a + b\,R_{\mathrm{i}} + c\,R_{\mathrm{i}}^2 \tag{7.48}$$

Likewise, at the grid points just outside this boundary (i+1,j) and (i+2,j), the velocity can be represented with Equations 7.49 and 7.50, respectively.

$$\begin{aligned} v_{\mathrm{i+1,j}} &= a + b\,(R_{\mathrm{i}} + \Delta r) + c\,(R_{\mathrm{i}} + \Delta r)^2 \\ &= a + b\,R_{\mathrm{i}} + b\,\Delta r + c\,R_{\mathrm{i}}^2 + c\,\Delta r^2 + 2\,c\,R_{\mathrm{i}}\,\Delta r \end{aligned} \tag{7.49}$$

$$\begin{aligned} v_{\mathrm{i+2,j}} &= a + b\,(R_{\mathrm{i}} + 2\,\Delta r) + c\,(R_{\mathrm{i}} + 2\,\Delta r)^2 \\ &= a + b\,R_{\mathrm{i}} + 2\,b\,\Delta r + c\,R_{\mathrm{i}}^2 + 4\,c\,\Delta r^2 + 4\,c\,R_{\mathrm{i}}\,\Delta r \end{aligned} \tag{7.50}$$

To produce some sensible result from the two above equations, the second one (Equation 7.50) is multiplied by -1 and the first one by $+4$ (Equation 7.49). By adding the two results together and then solving for b, gives the following:

$$b = \frac{4\,v_{\mathrm{i+1,j}} - v_{\mathrm{i+2,j}} - 3\,v_{\mathrm{i,j}}}{2\,\Delta r} - 2\,c\,R_{\mathrm{i}} \tag{7.51}$$

Finally, taking the derivative of $v_\theta = a + b\,r + c\,r^2$ at $r = R_{\mathrm{i}}$, with the concomitant use of the above equation, the algebraic counterpart of the derivative $\partial v_\theta / \partial r$ is produced:

$$\left.\frac{\partial v_\theta}{\partial r}\right|_{R_{\mathrm{i}}} = b + 2\,c\,R_{\mathrm{i}} = \frac{4\,v_{\mathrm{i+1,j}} - v_{\mathrm{i+2,j}} - 3\,v_{\mathrm{i,j}}}{2\,\Delta r} \tag{7.52}$$

The above is called **one-sided differences**, because it expresses a derivative of the velocity in terms of velocity values on only one side of that point.

What now remains, is to create the corresponding algebraic equation that applies to the right $r = R_{\mathrm{o}}$, in Figure 7.3. Creating a similar counterpart of Equation 7.48, the velocity $v_{\mathrm{i,j}}$ at $r = R_{\mathrm{o}}$ is given by $v_{\mathrm{i,j}} = a + b\,R_{\mathrm{o}} + c\,R_{\mathrm{o}}^2$. In the same fashion, the velocity at $r = R_{\mathrm{o}} - \Delta r$ and $r = R_{\mathrm{o}} - 2\,\Delta r$ (i.e. at the grid points (i-1,j) and (i-2,j)) is given by $v_{\mathrm{i-1,j}} = a + b\,(R_{\mathrm{o}} - \Delta r) + c\,(R_{\mathrm{o}} - \Delta r)^2$ and $v_{\mathrm{i-2,j}} = a + b\,(R_{\mathrm{o}} - 2\,\Delta r) +$

$c\,(R_\mathrm{o} - 2\,\Delta r)^2$, respectively. By going through the same procedure as was done to produce Equation 7.52, the algebraic counterpart of the differential $\partial v_\theta/\partial r$ at $r = R_\mathrm{o}$ becomes:

$$\left.\frac{\partial v_\theta}{\partial r}\right|_{R_\mathrm{o}} = \frac{-4\,v_{\mathrm{i}-1,\mathrm{j}} + v_{\mathrm{i}-2,\mathrm{j}} + 3\,v_{\mathrm{i,j}}}{2\,\Delta r} \tag{7.53}$$

The corresponding algebraic equation that applies on the top and bottom part of Figure 7.3 (i.e. at $z = H$ and $z = 0$), can be extracted directly from Equations 7.53 and 7.52, respectively:

$$\left.\frac{\partial v_\theta}{\partial z}\right|_{H} = \frac{-4\,v_{\mathrm{i,j}-1} + v_{\mathrm{i,j}-2} + 3\,v_{\mathrm{i,j}}}{2\,\Delta z} \quad \wedge \quad \left.\frac{\partial v_\theta}{\partial z}\right|_{0} = \frac{4\,v_{\mathrm{i,j}+1} - v_{\mathrm{i,j}+2} - 3\,v_{\mathrm{i,j}}}{2\,\Delta z} \tag{7.54}$$

Knowing that Neumann boundary condition at the top consists of $\partial v_\theta/\partial z = 0$, the above expression is equal to zero. Solving for $v_{\mathrm{i,j}}$, gives Equation 7.55.

$$\left.\frac{\partial v_\theta}{\partial z}\right|_{H} = 0 \quad \Rightarrow \quad v_{\mathrm{i,j}} = \frac{4\,v_{\mathrm{i,j}-1} - v_{\mathrm{i,j}-2}}{3} \tag{7.55}$$

With the help of Equation 7.9, it can be shown that Equations 7.52, 7.53 and 7.54 have the accuracy (or truncation error) of $O(\Delta r^2)$ and $O(\Delta z^2)$, respectively [2]. Finally, by using Equations 7.52 and 7.53, the algebraic presentation of the Robin boundary condition can be calculated as shown by the two following results:

$$\left[\frac{\partial v_\theta}{\partial r} = \frac{v_\theta}{r}\right]\Bigg|_{R_\mathrm{i}} \quad \Rightarrow \quad v_{\mathrm{i,j}} = \frac{4\,v_{\mathrm{i}+1,\mathrm{j}} - v_{\mathrm{i}+2,\mathrm{j}}}{3 + 2\,\Delta r/R_\mathrm{i}} \tag{7.56}$$

$$\left[\frac{\partial v_\theta}{\partial r} = \frac{v_\theta}{r}\right]\Bigg|_{R_\mathrm{o}} \quad \Rightarrow \quad v_{\mathrm{i,j}} = \frac{4\,v_{\mathrm{i}-1,\mathrm{j}} - v_{\mathrm{i}-2,\mathrm{j}}}{3 - 2\,\Delta r/R_\mathrm{o}} \tag{7.57}$$

Initial Condition

The initial condition $v_\theta(r, z, 0)$ for the transient problem is created by calculating the steady state case of the PDE with the boundary condition $v_\theta = v_0(r, 0)$. This is shown in the algorithm presented in Section 7.11.2.

7.8 Iteration Scheme

The iteration scheme used, when solving the time independent problem, is called **The Pseudotransient Method** [33]. It consists of gaining solution for the steady state problem, by time marching the transient Equation 7.6 (then with a constant boundary condition), until a steady state is reached (i.e. until $\partial v_\theta/\partial t$ is sufficiently close to zero). As such, the time plays the role of iteration parameter. For the full transient problem, the same approach is used, however with continuous changes in the boundary condition: $v_\theta(r, z, t) = v_0(r, t)\ \forall\ (r, z) \in \partial\Omega_\mathrm{D}$.

Combining Equations 7.6, 7.7 and 7.8, and keeping the discussion from Section 7.7 in mind, gives the overall problem to be solved, presented with the following PDE:

$$\begin{aligned}\rho\frac{\partial v_\theta}{\partial t} = {} & \frac{\partial}{\partial r}\left[\eta(v_\theta)\left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right)\right] + \frac{2\,\eta(v_\theta)}{r}\left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right) + \\ & + \frac{\partial}{\partial z}\left(\eta(v_\theta)\frac{\partial v_\theta}{\partial z}\right) \quad \forall \quad (r, z) \in \Omega\end{aligned} \tag{7.58}$$

and with the succeeding auxiliary conditions...

$$\begin{array}{rcccccc} v_\theta & = & v_\theta(r,z,0) & \forall & (r,z) & \in & \Omega \\ v_\theta & = & v_0(r,t) & \forall & (r,z) & \in & \partial\Omega_{\mathrm{D}} \\ \frac{\partial v_\theta}{\partial z} & = & 0 & \forall & (r,z) & \in & \partial\Omega_{\mathrm{N}} \\ \frac{\partial v_\theta}{\partial r} & = & \frac{v_\theta}{r} & \forall & (r,z) & \in & \partial\Omega_{\mathrm{R}} \end{array} \tag{7.59}$$

As mentioned previously, for steady state (i.e. time independence), the Dirichlet boundary condition consists of $v_0 = v_0(r)$. However, for the transient case (i.e. time dependence), this condition consists of $v_0 = v_0(r,t)$. In the above, the Neumann boundary condition (at $\partial\Omega_{\mathrm{N}}$) and the Robin boundary condition (at $\partial\Omega_{\mathrm{R}}$) will never occur simultaneously in a particular problem. In this work, the boundary condition consist of either Dirichlet and Neumann type, or of the Dirichlet and Robin type.

Equation 7.58 contains a term $\eta(v_\theta)$ which is a function of the primary unknown. Since this term is multiplied by the derivative of the primary unknown, this equation is a nonlinear PDE [66, 117]. As such one cannot write the discrete Equations 7.28 and 7.30 in the form of a linear system: $\mathbf{A}\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$, since the quantities A, B, C, D, E, and F (see Equations 7.32 to 7.37) are also dependent on the primary unknown $\mathbf{v}^{\mathrm{k}+1}$. Hence, the system of algebraic equations are nonlinear and are of the form $\mathbf{A}(\mathbf{v}^{\mathrm{k}+1})\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$. To come around this problem, a so-called **Successive Substitution Method** [66] is used. In basic terms, this method consists of converting the nonlinear system $\mathbf{A}(\mathbf{v}^{\mathrm{k}+1})\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$ into a sequence of linear ones $\mathbf{A}\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$. A short introduction to this method will now follow, but further information about it can be found in a textbook by Langtangen [66].

For the given time step $\mathrm{k}+1$, the quest is to gain the solution $\mathbf{v}^{\mathrm{k}+1}$. The idea consists of starting with some guess of this solution, called $\mathbf{v}_0$. Here, it is always the solution from the previous time step k that is used as a first guess $\mathbf{v}_0 \leftarrow \mathbf{v}^{\mathrm{k}}$. Inserting $\mathbf{v}_0$ into the terms A, B, C, D, E, and F converts the array $\mathbf{A}(\mathbf{v}^{\mathrm{k}+1})$ into a known array $\mathbf{A}(\mathbf{v}_0)$. This means that Equations 7.28 and 7.30 become a system of linear equations: $\mathbf{A}(\mathbf{v}_0)\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$. The solution of this equation can be named $\mathbf{v}_1$. Of course, the solution $\mathbf{v}_1$ is not the correct solution $\mathbf{v}^{\mathrm{k}+1}$ of Equations 7.28 and 7.30, since the array $\mathbf{A}(\mathbf{v}_0)$ was wrong to begin with. Hopefully, $\mathbf{v}_1$ is a better approximation than $\mathbf{v}_0$ to the exact solution. The next step consists of inserting $\mathbf{v}_1$ into the corresponding array $\mathbf{A}$ and then solving the system $\mathbf{A}(\mathbf{v}_1)\cdot\mathbf{v}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}})$ all over again to produce $\mathbf{v}_2$. The successive substitution method consists of repeating this process until $\|\mathbf{v}_{\mathrm{s}} - \mathbf{v}_{\mathrm{s}-1}\|_{\mathbf{rms}} \leq \mathrm{tol}$ where the term "tol" is an acceptable difference between two successive solutions (see Equations 7.73 and 7.74). The whole idea can be summarized in the following iteration scheme:

$$\mathbf{A}(\mathbf{v}_{\mathrm{s}-1}^{\mathrm{k}+1})\cdot\mathbf{v}_{\mathrm{s}}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}^{\mathrm{k}}) \quad \wedge \quad \mathbf{v}_0^{\mathrm{k}+1} = \mathbf{v}^{\mathrm{k}} \quad ; \quad \mathrm{s} = 1,2,3,\ldots \tag{7.60}$$

Note that the above iteration is for a given (constant) time step k+1 and therefore the iteration index **s** should not be confused with the time index **k**. In the algorithm used in this work (see Section 7.11), the above equation is presented as:

$$\mathbf{A}(\mathbf{v}_{\mathrm{i,j}}^{\mathrm{k}+1})\cdot\mathbf{v}_{\mathrm{i,j;NEW}}^{\mathrm{k}+1} = \mathbf{b}(\mathbf{v}_{\mathrm{i,j}}^{\mathrm{k}}) \tag{7.61}$$

The term $\mathbf{v}_{\mathrm{i,j}}^{\mathrm{k}+1}$, could equally have been represented with $\mathbf{v}_{\mathrm{i,j;OLD}}^{\mathrm{k}+1}$ in the above. Since the PDE to be solved can be highly nonlinear due to the shear viscosity $\eta(r,z,t)$, the successive substitution method alone may diverge. Therefor the **Continuation Method** [66] is also included into the main source code **main.f90**. The continuation

parameter Λ is included into the shear viscosity in a such way that $\Lambda = 0$ gives a constant shear viscosity, as applies for Newtonian fluid $\eta =$ constant. For example, in the case of Bingham fluid, the shear viscosity is now rewritten to $\eta = \mu + \tau_{\mathrm{o}}^{\Lambda}/\dot{\gamma}$ or $\eta = \mu + \Lambda\,\tau_{\mathrm{o}}/\dot{\gamma}$, where $\Lambda = 1$ recovers the original problem. The general idea is to increase Λ in steps, from zero to unity (or in the case of Bingham fluid, slowly increasing the yield value), using the solution to the most recent Λ values as the start for a new nonlinear iteration. The disadvantage of this method is that it can only be used when solving a steady state problem.

In this work, the experience indicated that stability was best dealt with by reducing the time step Δt. Therefore all the numerical computations are done in two continuation steps only, starting by solving the Newtonian case ($\Lambda = 0$) and then using it as an initial guess to the original problem ($\Lambda = 1$). However, before starting with the first continuation step $\Lambda = 0$, a linear interpolation between Dirichlet boundary values over the solution area is done to create an initial guess for the Newtonian case. The overall solution algorithm is shown in Section 7.11.

7.9 The Regularization Parameter

The fluid in this work consists of a viscoplastic material. As such it can be in a two different states, namely the viscoplastic state (Ω_{p}) and the solid state (Ω_{e}). These two conditions are discussed in Section 3.2 (Page 52) and are best described with the von Mises yield condition as shown with Equation 7.62.

$$\boldsymbol{\sigma} = \begin{cases} \boldsymbol{\sigma}^{\mathrm{P}} & \text{if} \quad -\mathrm{II}_{\mathbf{S}}^{\mathrm{E}} \geq C_{\mathrm{y}} \\ \boldsymbol{\sigma}^{\mathrm{E}} & \text{if} \quad -\mathrm{II}_{\mathbf{S}}^{\mathrm{P}} < C_{\mathrm{y}} \end{cases} \tag{7.62}$$

In this thesis, the zone which is in a solid state ($\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{E}}$), is designated with Ω_{e}, while the zone of viscoplastic state ($\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{P}}$), is designated by Ω_{p} and hence $\Omega \in [\Omega_{\mathrm{e}} \cup \Omega_{\mathrm{p}}]$.

In the above equation, the only assumption made for the stress tensor $\boldsymbol{\sigma}^{\mathrm{P}}$, is that it depends at least on the strain rate tensor $\dot{\boldsymbol{\varepsilon}}$. For example, a pure Newtonian fluid can be assumed: $\boldsymbol{\sigma}^{\mathrm{P}} = -p\,\mathbf{I} + 2\,\mu\,\dot{\boldsymbol{\varepsilon}}$, if suitable for the test material in question. In this thesis, the stress tensor consist of $\boldsymbol{\sigma}^{\mathrm{P}} = -p\,\mathbf{I} + 2\,\eta\,\dot{\boldsymbol{\varepsilon}}$, where the shear viscosity is on the form $\eta = B(\dot{\gamma}, \tilde{\Gamma}, \tilde{\Theta}, t, \dots) + [C_{\mathrm{y}}(\dot{\gamma}, \tilde{\Gamma}, \tilde{\Theta}, t, \dots)]^{\frac{1}{2}}/\dot{\gamma}$. Fortunately, the shear viscosity η depends on the shear rate $\dot{\gamma}$ in such manner that $\lim_{\dot{\gamma}\to 0}\eta = \infty$. With this characteristic, the stress tensor $\boldsymbol{\sigma}^{\mathrm{P}}$ can be used in imitating the solid state. The justification for this is more physical than mathematical: For example, *when looking at a window of glass, the immediate perception is that it presents a solid. In waiting half a century or so, it becomes apparent that it deforms and flows downward under the action of gravity* $\mathbf{g}$*. When modeling such event, an extremely large shear viscosity* η *must be used, which results in the extremely low shear rate* $\dot{\gamma}$ *observed. With a stiffer (or more rigid) glass, a still larger shear viscosity must be applied.* In generalizing this concept, the solid can be considered as fluid with a very large shear viscosity value η, deforming under extremely low shear rate $\dot{\gamma} \approx 0$. A perfect (or fully rigid) solid would then have infinite[3] shear viscosity $\eta \to \infty$ and zero shear rate $\dot{\gamma} = 0$. This specific characteristics automatically applies for the shear viscosity used in this work, since $\lim_{\dot{\gamma}\to 0}\eta = \infty$.

[3]The perception that a perfect solid (or a rigid body movement of a fluid) consist of $\eta \to \infty$ is also made by Langtangen [66], when defining the Herschel-Bulkley model.

However, the perfect solid state cannot be calculated by the means of fluid mechanics. In trying to do so, leads to an incorrect result. For example, assuming the test material consist of Bingham fluid, gives the shear stress of $T_{r\theta}(r) = \eta\,(dv_\theta/dr - v_\theta/r) = \eta\,\dot{\gamma} = \mu\,\dot{\gamma} + \tau_\mathrm{o}$ (see Equation 3.12 on Page 57). Furthermore, according to Equation 3.16 the torque within the test sample is $\hat{T} = 2\,\pi\,r^2\,h\,T_{r\theta} = 2\,\pi\,r^2\,h\,(\mu\,\dot{\gamma} + \tau_\mathrm{o})$. Trying to utilize this result in $r \in [R_\mathrm{s}, R_\mathrm{o}]$ (i.e. $r \in \Omega_\mathrm{e}$) gives $\hat{T}|_{\dot{\gamma}=0} = 2\,\pi\,r^2\,h\,\tau_\mathrm{o} \propto r^2 \neq$ constant, which violates the condition described with Equation 3.15.

To calculate the solid state by means of fluid mechanics, a small modification must be applied to shear viscosity. This consists of adding a small **regularization parameter** δ [13, 130] to the shear rate: $\eta = B(\dot{\gamma}, \tilde{\Gamma}, \tilde{\Theta}, t, \dots) + [C_\mathrm{y}(\dot{\gamma}, \tilde{\Gamma}, \tilde{\Theta}, t, \dots)]^{\frac{1}{2}}/(\dot{\gamma}+\delta)$. The objectives with this operation is to inhibit the possibility of infinite shear viscosity $\eta \not\to \infty$ and hence inhibit zero shear rate $\dot{\gamma} \neq 0$. As such, the solid state no longer consist of the condition $\dot{\gamma} = 0$, but rather of $\dot{\gamma} \approx 0$. This is the basic role the regularization parameter δ, namely to produce a nonzero and very small shear rate ($\dot{\gamma} \approx 0$) in the solid domain Ω_e, to imitate the prefect solid state ($\dot{\gamma} = 0$).

Numerical results produced in this work, give that the smallest shear rate value $\dot{\gamma}_\mathrm{min}$ in this zone, is often much smaller than the regularization parameter δ used. Hence δ presents by no means the minimum shear rate allowed. However, as shown with Figure 8.11 (Page 193), the general characteristics is that the smaller δ is, the smaller the shear rate $\dot{\gamma}$ becomes in the solid domain Ω_e.

Strictly speaking, the overall numerical problem consists of solving Equation 7.58 in the viscoplastic domain, namely in Ω_p. The solution in the solid zone Ω_e is always known and consist of velocity for a rotating solid $v_\theta(r, z, t) = r\,\omega_\mathrm{o}(t)$ and of zero shear rate $\dot{\gamma} = 0$. Therefore, the boundary condition used between the two domains Ω_e and Ω_p, must consist of the Dirichlet boundary condition $v_\theta = r\,\omega_\mathrm{o}$. Since the location of this boundary is unknown prior to the numerical calculation, it becomes difficult to apply the Dirichlet condition at the correct grid points. To overcome this dilemma, the regularization parameter is applied: When solving Equation 7.58 with the shear viscosity of $\eta = B(\dot{\gamma}, \tilde{\Gamma}, \tilde{\Theta}, t, \dots) + [C_\mathrm{y}(\dot{\gamma}, \tilde{\Gamma}, \tilde{\Theta}, t, \dots)]^{\frac{1}{2}}/(\dot{\gamma} + \delta)$, the test material consists of a non-viscoplastic fluid that can imitate both a viscoplastic state and a solid state. As such, the governing Equation 7.58 is rather solved over the whole solution region $\Omega \in \Omega_\mathrm{p} \cup \Omega_\mathrm{e}$. Hence, the symbols Ω_e and Ω_p will now designate the domain of the imitated solid state and the imitated viscoplastic state, respectively. From here on, these two imitated states will be referred simply as the "solid state" and the "viscoplastic state". The validation of the overall above approach is tested in Sections 8.5 and 8.6.

Since the computer and the solution algorithm have difficulty in working with large values of shear viscosity η and large changes in its value $\Delta\eta$, the regularization parameter δ is usually kept between $10^{-3}\,\mathrm{s}^{-1}$ and $5 \cdot 10^{-3}\,\mathrm{s}^{-1}$. Using a smaller regularization parameter requires a smaller time step Δt (and hence a longer computation time) in order to maintain numerical stability. As shown in Sections 8.5 and 8.6, the larger this parameter is, the more incorrect the solution becomes, i.e. a worse imitation of the solid and viscoplastic state is produced.

7.10 Shear Viscosity with Fading Memory

In Chapter 9, a complex thixotropic analysis is made for some cement pastes using specific types of lignosulfonates. Basically, the analysis consists of reproducing measured torque $\hat{T}$ by numerical means, and in doing so, extracting a more detailed

shear viscosity function. To make this possible, it was necessary to introduce a fading memory into the shear viscosity function $\eta = \eta(\dot{\gamma}, U_3, t)$, where $U_3 = U_3(\tilde{\Gamma}, \tilde{\Theta}, t)$ describes the current (reversible) coagulation state of the cement particles (see Section 2.4.2). It is the terms $\tilde{\Gamma}$ and $\tilde{\Theta}$ that are the memory modules. They are defined with Equations 7.63 and 7.64, receptively.

$$\tilde{\Gamma}(r, z, t) = \int_0^t \alpha(t - t')\ \dot{\gamma}(r, z, t')\ dt' \tag{7.63}$$

$$\tilde{\Theta}(r, z, t) = \int_0^t \beta(t - t')\ H(\dot{\gamma}, r, z, t')\ dt' \tag{7.64}$$

The terms α and β are memory functions. The variable H is the coagulation rate coefficient, described in Sections 2.4.2 and 2.5.3. For a programming point of view, it is very important to be able to split the memory functions α and β into two separate components, as shown below:

$$\alpha(t - t') = e^{-(t-t')/m_{\mathrm{a}}} = e^{-t/m_{\mathrm{a}}}\, e^{t'/m_{\mathrm{a}}} = \alpha_{\mathrm{II}}(t)\, \alpha_{\mathrm{I}}(t') \tag{7.65}$$

$$\beta(t - t') = e^{-(t-t')/m_{\mathrm{b}}} = e^{-t/m_{\mathrm{b}}}\, e^{t'/m_{\mathrm{b}}} = \beta_{\mathrm{II}}(t)\, \beta_{\mathrm{I}}(t') \tag{7.66}$$

As is described in Sections 2.4.2, the term $U_3|_{t=0} = U_{\mathrm{o}}$ describes the (reversible) coagulation state of the cement particles, at the beginning of experimental observation. In this thesis, the start always occurs at the time $t = 0$ and ends at $t = 50\,\mathrm{s}$. This time is not to be confused with the time duration from mixing of water and cement particles. The former time period spans only 50 seconds (see Figure 9.1, to the left), while the latter spans the whole 102 minutes (see Section 9.2.3). The initial condition U_{o} is dependent on the past history of shear rate and on the past history of coagulation rate. That is, U_{o} depends on the memory modules $\tilde{\Gamma}$ and $\tilde{\Theta}$ when the integration is from $t' = -\infty$ to $t' = 0$. Therefore, when calculating a new condition $U_3(t)$ for the given initial condition $U_3(0) = U_{\mathrm{o}}$, the integration must begin from the start of an experiment $t' = 0$ to the current time $t' = t$ (where $t \in [0, 50\,\mathrm{s}]$) as is shown with Equations 7.63 and 7.64.

The discrete counterpart of the terms $\tilde{\Gamma}(r, z, t)$ and $\tilde{\Theta}(r, z, t)$ are presented with $\tilde{\Gamma}^{\mathrm{k}}_{\mathrm{i,j}}$ and $\tilde{\Theta}^{\mathrm{k}}_{\mathrm{i,j}}$, and are given by Equations 7.67 and 7.68.

$$\tilde{\Gamma}^{\mathrm{k}}_{\mathrm{i,j}} = \alpha_{\mathrm{II}}(\mathrm{k}\Delta t) \sum_{\mathrm{k}'=0}^{\mathrm{k}} \alpha_{\mathrm{I}}(\mathrm{k}'\Delta t)\dot{\gamma}^{\mathrm{k}'}_{\mathrm{i,j}}\ \Delta t = \alpha_{\mathrm{II}}(\mathrm{k}\Delta t)\ \mathrm{FMSR}^{\mathrm{k}}_{\mathrm{i,j}} \tag{7.67}$$

$$\tilde{\Theta}^{\mathrm{k}}_{\mathrm{i,j}} = \beta_{\mathrm{II}}(\mathrm{k}\Delta t) \sum_{\mathrm{k}'=0}^{\mathrm{k}} \beta_{\mathrm{I}}(\mathrm{k}'\Delta t) H(\dot{\gamma}^{\mathrm{k}'}_{\mathrm{i,j}})\ \Delta t = \beta_{\mathrm{II}}(\mathrm{k}\Delta t)\ \mathrm{FMCR}^{\mathrm{k}}_{\mathrm{i,j}} \tag{7.68}$$

In the above, the term FMSR stands for **F**ading **M**emory **S**hear **R**ate module, while FMCR for **F**ading **M**emory **C**oagulation **R**ate module. They are both cumulative in nature, and resides in the source code **main.f90**.

From Equation 7.67, it is apparent that the condition $\tilde{\Gamma}(r, z, 0) = 0$ applies, since the integration from zero to zero, yield always a zero. This means that $\tilde{\Gamma}^{0}_{\mathrm{i,j}}$ must also be equal to zero, which is enforced by setting $\Delta t = 0$ at $\mathrm{k} = 0$. This approach is

self evident, since no time advancements Δt exists at $t = 0$ (i.e. at $\mathrm{k} = 0$). With the above in mind, the condition of $\mathrm{FMSR}^{0}_{\mathrm{i,j}} = 0$ must also be valid. The same type of consideration, as in the above, apply for the terms $\tilde{\Theta}(r, z, 0)$, $\tilde{\Theta}^{0}_{\mathrm{i,j}}$ and $\mathrm{FMCR}^{0}_{\mathrm{i,j}}$.

The discrete counterpart of the terms $\tilde{\Gamma}(r, z, t)$ and $\tilde{\Theta}(r, z, t)$ at the time step $\mathrm{k}+\frac{1}{2}$ and $\mathrm{k}+1$ is calculated according to Equations 7.69 to 7.72.

$$\tilde{\Gamma}^{\mathrm{k}+\frac{1}{2}}_{\mathrm{i,j}} = \alpha_{\mathrm{II}}((\mathrm{k}+1/2)\Delta\mathrm{t}) \left[\mathrm{FMSR}^{\mathrm{k}}_{\mathrm{i,j}} + \alpha_{\mathrm{I}}((\mathrm{k}+1/2)\Delta\mathrm{t})\, \dot{\gamma}^{\mathrm{k}+\frac{1}{2}}_{\mathrm{i,j}}\, \frac{\Delta\mathrm{t}}{2}\right] \tag{7.69}$$

$$\tilde{\Gamma}^{\mathrm{k}+1}_{\mathrm{i,j}} = \alpha_{\mathrm{II}}((\mathrm{k}+1)\Delta\mathrm{t}) \left[\mathrm{FMSR}^{\mathrm{k}}_{\mathrm{i,j}} + \alpha_{\mathrm{I}}((\mathrm{k}+1)\Delta\mathrm{t})\, \dot{\gamma}^{\mathrm{k}+1}_{\mathrm{i,j}}\, \Delta\mathrm{t}\right] \tag{7.70}$$

$$\tilde{\Theta}^{\mathrm{k}+\frac{1}{2}}_{\mathrm{i,j}} = \beta_{\mathrm{II}}((\mathrm{k}+1/2)\Delta\mathrm{t}) \left[\mathrm{FMCR}^{\mathrm{k}}_{\mathrm{i,j}} + \beta_{\mathrm{I}}((\mathrm{k}+1/2)\Delta\mathrm{t})\, \mathrm{H}(\dot{\gamma}^{\mathrm{k}+\frac{1}{2}}_{\mathrm{i,j}})\, \frac{\Delta\mathrm{t}}{2}\right] \tag{7.71}$$

$$\tilde{\Theta}^{\mathrm{k}+1}_{\mathrm{i,j}} = \beta_{\mathrm{II}}((\mathrm{k}+1)\Delta\mathrm{t}) \left[\mathrm{FMCR}^{\mathrm{k}}_{\mathrm{i,j}} + \beta_{\mathrm{I}}((\mathrm{k}+1)\Delta\mathrm{t})\, \mathrm{H}(\dot{\gamma}^{\mathrm{k}+1}_{\mathrm{i,j}})\, \Delta\mathrm{t}\right] \tag{7.72}$$

7.11 Algorithms

The algorithms for the steady state- and the transient flow calculation are shown in Sections 7.11.1 and 7.11.2, respectively. In both algorithms, the term "tol" is used to determine the termination of a successive substitution. As shown by Equation 7.73, depending on the continuation parameter λ, this term can have two different values.

$$\mathrm{tol} = \begin{cases} \mathrm{tol\_Newton} & \text{if} \quad \lambda = 0 \\ \mathrm{tol\_Plastic} & \text{if} \quad \lambda > 0 \end{cases} \tag{7.73}$$

The term "tol_Plastic" needs always to be much smaller than "tol_Newton", because of the much smaller difference between two successive steps in the former case. This is so since smaller time step is required when calculating the viscoplastic flow, in order to gain stability. The difference between two successive steps is monitored through their root mean square difference as shown in Equation 7.74.

$$\|v^{\mathrm{k}+1}_{\mathrm{i,j}} - v^{\mathrm{k}+1}_{\mathrm{i,j;NEW}}\|_{\mathbf{rms}} \equiv \sqrt{\frac{1}{\mathrm{N}} \sum_{(i,j)\,\in\,\Omega} \left[\frac{v^{\mathrm{k}+1}_{\mathrm{i,j}} - v^{\mathrm{k}+1}_{\mathrm{i,j;NEW}}}{\frac{1}{2}(v^{\mathrm{k}+1}_{\mathrm{i,j}} + v^{\mathrm{k}+1}_{\mathrm{i,j;NEW}})}\right]^2} \tag{7.74}$$

In the above, the term N presents the number of grid points in the solution area Ω. When the condition "$\|v^{\mathrm{k}+1}_{\mathrm{i,j}} - v^{\mathrm{k}+1}_{\mathrm{i,j;NEW}}\|_{\mathbf{rms}} < \mathrm{tol}$" becomes valid, the calculation of the next time step $\mathrm{k}+2$ is activated.

The above discussions apply for both the time independent and the transient case. However, when calculating a time independent problem, an additional restriction is required, namely the condition when determining a time independence. The term "tol_RMS_active" is used in relation when determining time independence and hence is being part of stopping calculations when appropriate. This variable can have two values, given by Equation 7.75.

$$\mathrm{tol\_RMS\_active} = \begin{cases} \mathrm{tol\_Newton} & \text{if} \quad \lambda = 0 \\ \mathrm{tol\_RMS} & \text{if} \quad \lambda > 0 \end{cases} \tag{7.75}$$

The terms "tol_Newton" and "tol_RMS" are set by the user. After each time step, the condition "$\|v_{\mathrm{i,j}}^{\mathrm{k}} - v_{\mathrm{i,j}}^{\mathrm{k+1}}\|_{\mathbf{rms}} < \mathrm{tol\_RMS\_active}$" is checked if valid (see Equation 7.76). If so, the calculations are stopped and all results are written into different files.

$$\|v_{\mathrm{i,j}}^{\mathrm{k}} - v_{\mathrm{i,j}}^{\mathrm{k+1}}\|_{\mathbf{rms}} \equiv \sqrt{\frac{1}{\mathrm{N}} \sum_{(i,j)\,\in\,\Omega} \left[\frac{v_{\mathrm{i,j}}^{\mathrm{k}} - v_{\mathrm{i,j}}^{\mathrm{k+1}}}{\frac{1}{2}(v_{\mathrm{i,j}}^{\mathrm{k}} + v_{\mathrm{i,j}}^{\mathrm{k+1}})} \right]^2} \tag{7.76}$$

Sometimes, the term tol_RMS_active can be accidentally set to be too low, which leads to that the condition "$\|v_{\mathrm{i,j}}^{\mathrm{k}} - v_{\mathrm{i,j}}^{\mathrm{k+1}}\|_{\mathbf{rms}} < \mathrm{tol\_RMS\_active}$" becomes never valid. The term "ZERO_TIME" is used in avoiding such a situation. It determines the maximum amount (pseudo) time iterations to be calculated. For example, using $\delta t = 10^{-6}\,\mathrm{s}$ and $\mathrm{ZERO\_TIME} = 2\,\mathrm{s}$, leads to $2 \cdot 10^6$ (pseudo) time iterations, at the most. This method of termination is not shown in the algorithm of Section 7.11.1.

7.11.1 Algorithm of Time Independent Code

Algorithm 1: Steady State Problem

Main variables: $v_{\mathrm{i,j}}^{\mathrm{k}}$, $v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}$, $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+\frac{1}{2}}$,$v_{\mathrm{i,j}}^{\mathrm{k}+1}$, $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$
Setting up linear approximation for $v_{\mathrm{i,j}}^{\mathrm{k}}$ at the time step k=0
Continuation method [66]:
do $\lambda = 0, \ldots 1$ (CONTINUATION: DO ...)
  $\lambda = 0 \Rightarrow$ Newtonian fluid $\wedge$ $\lambda > 0 \Rightarrow$ non-Newtonian fluid
  Pseudotransient method [33]:
  while $\|v_{\mathrm{i,j}}^{\mathrm{k}} - v_{\mathrm{ij}}^{\mathrm{k}+1}\|_{\mathbf{rms}} >$ tol_RMS_active **do** (ZERO_TIME_LOOP: DO WHILE ...)
    convergence = $false$
    $v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \leftarrow v_{\mathrm{i,j}}^{\mathrm{k}} \wedge v_{\mathrm{i,j}}^{\mathrm{k}+1} \leftarrow v_{\mathrm{i,j}}^{\mathrm{k}}$
    Successive substitution [66]:
    while not convergence, do (CONVERGE: DO WHILE ...)
      do j = 1,NY2
        do i = 1,NX2 (CALL MATRIX_UPDATE_X(...))
          Calculate $\dot{\gamma}^{\mathrm{k}} = \dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}})$ and $\dot{\gamma}^{\mathrm{k}+\frac{1}{2}} = \dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}})$ from Equations 7.38 to 7.42. Calculate $\eta^{\mathrm{k}} = \eta(\lambda, \dot{\gamma}^{\mathrm{k}}, \mathrm{k}\ \Delta t)$ and $\eta^{\mathrm{k}+\frac{1}{2}} = \eta(\lambda, \dot{\gamma}^{\mathrm{k}+\frac{1}{2}}, (\mathrm{k}+\frac{1}{2})\Delta t)$. Calculate A, B, C for the time step $\mathrm{k}+\frac{1}{2}$ (Equations 7.32 to 7.34). Calculate D, E, F (Equations 7.35 to 7.37) and thereafter K (Equation 7.29) for the time step k.
        end do
        Solve Equation 7.28 to calculate $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+\frac{1}{2}}$ (CALL MATRIX_SOLVER(...))
      end do
      $v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \leftarrow v_{\mathrm{i,j;NEW}}^{\mathrm{k}+\frac{1}{2}}$
      do i = 1,NX2
        do j = 1,NY2 (CALL MATRIX_UPDATE_Y(...))
          Calculate $\dot{\gamma}^{\mathrm{k}+\frac{1}{2}} = \dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}})$ and $\dot{\gamma}^{\mathrm{k}+1} = \dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}+1})$ from Equations 7.38 to 7.42. Calculate $\eta^{\mathrm{k}+\frac{1}{2}} = \eta(\lambda, \dot{\gamma}^{\mathrm{k}+\frac{1}{2}}, (\mathrm{k}+\frac{1}{2})\Delta t)$ and $\eta^{\mathrm{k}+1} = \eta(\lambda, \dot{\gamma}^{\mathrm{k}+1}, (\mathrm{k}+1)\Delta t)$. Calculate A, B, C (Equations 7.32 to 7.34) and L (Equation 7.31) for the time step $\mathrm{k}+\frac{1}{2}$. Calculate D, E, F (Equations 7.35 to 7.37) for the time step $\mathrm{k}+1$.
        end do
        Solve Equation 7.30 to calculate $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$ (CALL MATRIX_SOLVER(...))
      end do
      if $\|v_{\mathrm{i,j}}^{\mathrm{k}+1} - v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}\|_{\mathbf{rms}} \leq$ tol **then** $\Rightarrow$ **convergence** = $true$
      $v_{\mathrm{i,j}}^{\mathrm{k}+1} \leftarrow v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$
    end while (END DO CONVERGE)
    $v_{\mathrm{i,j}}^{\mathrm{k}} \leftarrow v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$
    $\mathrm{k} \leftarrow \mathrm{k}+1$
  end while (END DO ZERO_TIME_LOOP)
end do (END DO CONTINUATION)

7.11.2 Algorithm of Time Dependent Code

Algorithm 2: Transient Problem

Main variables: $v_{\mathrm{i,j}}^{\mathrm{k}}$, $v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}$, $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+\frac{1}{2}}$,$v_{\mathrm{i,j}}^{\mathrm{k}+1}$, $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$. The result from **Algorithm 1** is used as an initial condition for $v_{\mathrm{i,j}}^{\mathrm{k}}$ at the time step $\mathrm{k}=0$ (with the boundary condition $\omega_{\mathrm{o}}=\omega_{\mathrm{o}}(0)$). Initial memory: $\tilde{\Gamma}_{\mathrm{i,j}}^{0}=\mathrm{FMSR}_{\mathrm{i,j}}^{0}=0 \wedge \tilde{\Theta}_{\mathrm{i,j}}^{0}=\mathrm{FMCR}_{\mathrm{i,j}}^{0}=0$.

do $\mathrm{k}=0,\mathrm{N_{dt}}$ (TIME_LOOP: DO k = 0, N_dt)

convergence = *false*

$v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \leftarrow v_{\mathrm{i,j}}^{\mathrm{k}}$ and update boundary condition $\omega_{\mathrm{o}}=\omega_{\mathrm{o}}((\mathrm{k}+\frac{1}{2})\Delta t)$

$v_{\mathrm{i,j}}^{\mathrm{k}+1} \leftarrow v_{\mathrm{i,j}}^{\mathrm{k}}$ and update boundary condition $\omega_{\mathrm{o}}=\omega_{\mathrm{o}}((\mathrm{k}+1)\Delta t)$

Successive substitution [66]:

while not convergence, do (TIME_CONVERGE: DO WHILE ...)

do j = 1,NY2

do i = 1,NX2 (CALL MATRIX_UPDATE_X(...))

1) $\dot{\gamma}^{\mathrm{k}}=\dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}}) \wedge \dot{\gamma}^{\mathrm{k}+\frac{1}{2}}=\dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}) \leftarrow$ Equations 7.38 to 7.42

2) $\tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}} \wedge \tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \wedge \tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}} \wedge \tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \leftarrow$ Equations 7.67, 7.69, 7.68 and 7.71

3) $\eta^{\mathrm{k}}=\eta(\dot{\gamma}^{\mathrm{k}},\tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}},\tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}},\mathrm{k}\Delta t) \wedge \eta^{\mathrm{k}+\frac{1}{2}}=\eta(\dot{\gamma}^{\mathrm{k}+\frac{1}{2}},\tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}},\tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}},(\mathrm{k}+\frac{1}{2})\Delta t)$

4) Calculate A, B, C for the time step $\mathrm{k}+\frac{1}{2}$ (Equations 7.32 to 7.34). Calculate D, E, F (Equations 7.35 to 7.37) and thereafter K (Equation 7.29) for the time step k.

end do

Solve Equation 7.28 to calculate $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+\frac{1}{2}}$ (CALL MATRIX_SOLVER(...))

end do

$v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \leftarrow v_{\mathrm{i,j;NEW}}^{\mathrm{k}+\frac{1}{2}}$

do i = 1,NX2

do j = 1,NY2 (CALL MATRIX_UPDATE_Y(...))

1) $\dot{\gamma}^{\mathrm{k}+\frac{1}{2}}=\dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}}) \wedge \dot{\gamma}^{\mathrm{k}+1}=\dot{\gamma}(v_{\mathrm{i,j}}^{\mathrm{k}+1}) \leftarrow$ Equations 7.38 to 7.42

2) $\tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \wedge \tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}+1} \wedge \tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}} \wedge \tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}+1} \leftarrow$ Equations 7.69 to 7.72

3) $\eta^{\mathrm{k}+\frac{1}{2}}=\eta(\dot{\gamma}^{\mathrm{k}+\frac{1}{2}},\tilde{\Gamma}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}},\tilde{\Theta}_{\mathrm{i,j}}^{\mathrm{k}+\frac{1}{2}},(\mathrm{k}+\frac{1}{2})\Delta t) \wedge \eta^{\mathrm{k}+1}=\cdots$

4) Calculate A, B, C (Equations 7.32 to 7.34) and L (Equation 7.31) for the time step $\mathrm{k}+\frac{1}{2}$. Calculate D, E, F (Equations 7.35 to 7.37) for the time step $\mathrm{k}+1$.

end do

Solve Equation 7.30 to calculate $v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$ (CALL MATRIX_SOLVER(...))

end do

if $\|v_{\mathrm{i,j}}^{\mathrm{k}+1}-v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}\|_{\mathbf{rms}} \leq \mathrm{tol}$ **then** $\Rightarrow$ **convergence** = *true*

$v_{\mathrm{i,j}}^{\mathrm{k}+1} \leftarrow v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1}$

end while (END DO TIME_CONVERGE)

$\mathrm{FMSR}_{\mathrm{i,j}}^{\mathrm{k}+1}=\mathrm{FMSR}_{\mathrm{i,j}}^{\mathrm{k}}+\alpha_{\mathrm{I}}((\mathrm{k}+1)\Delta t)\,\dot{\gamma}_{\mathrm{i,j}}^{\mathrm{k}+1}\,\Delta t$

$\mathrm{FMCR}_{\mathrm{i,j}}^{\mathrm{k}+1}=\mathrm{FMCR}_{\mathrm{i,j}}^{\mathrm{k}}+\beta_{\mathrm{I}}((\mathrm{k}+1)\Delta t)\,H(\dot{\gamma}_{\mathrm{i,j}}^{\mathrm{k}+1})\,\Delta t$

$v_{\mathrm{i,j}}^{\mathrm{k}} \leftarrow v_{\mathrm{i,j;NEW}}^{\mathrm{k}+1} \wedge \mathrm{FMSR}_{\mathrm{i,j}}^{\mathrm{k}} \leftarrow \mathrm{FMSR}_{\mathrm{i,j}}^{\mathrm{k}+1} \wedge \mathrm{FMCR}_{\mathrm{i,j}}^{\mathrm{k}} \leftarrow \mathrm{FMCR}_{\mathrm{i,j}}^{\mathrm{k}+1}$

end do (END DO TIME_LOOP)

Chapter 8

Simulation of the ConTec Viscometers

8.1 Introduction

This chapter presents numerical simulations of ConTec **BML** Viscometer 3 and ConTec Viscometer 4 based on the theories presented in Chapter 7. The objectives of this chapter are mainly twofold. The first objective is to demonstrate the accuracy of the numerical code, both as a function of grid resolution used and also as a function of different magnitude of regularization parameter δ utilized. Issues like convergence, numerical convergence, apparent stability and stability are dealt with here.

The second objective is to present some numerical results made, to investigate how much z-dependency exists in the velocity function $v_\theta(r, z, t)$ in the upper part of the viscometer. This analysis has to do with justifying the assumption made in Section 3.3.1 (Page 56) when deriving the well-known Reiner-Riwlin equation.

For the first objective, the geometry of the ConTec **BML** Viscometer 3 is used. However for the second objective, both geometries are applied.

8.2 Source Code

Two softwares are presented in Appendix A. The first one is used when calculating the viscoplastic flow inside both ConTec viscometers. The second software is used in calculating the flow inside the C^3P^2-geometry (see Section 10.3.7). Both softwares will have the same name, namely Viscometric-ViscoPlastic-Flow (or VVPF). The reason for this is that both of them use much of the same subroutines, stored as modules. This is done to reduce developing time, when writing new software for an additional type of viscometer. The source codes are written in accordance with the **Fortran 90** standard (ANSI X3.198-1992 $\wedge$ ISO/IEC 1539-1:1991 (E)) and as such, can be used on any computer platform having a Fortran 90 compiler. These files are shown in Appendix A. Viscometric-ViscoPlastic-Flow, is free software; which can be redistributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at the users option) any later version. Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;

without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

When calculating the viscoplastic flow inside the ConTec viscometers, the software Viscometric-ViscoPlastic-Flow consists of seven files listed below.

1. **param.f90** (MODULE): This code defines and sets all variables of relevance, like R_i, R_o, h, Δr, Δz, Δt, tol, tol_RMS, f_min, f_max and so forth.

2. **motion.f90** (MODULE): This file reads the basic information from **param.f90** to produce angular velocity $\omega_\mathrm{o} = \omega_\mathrm{o}(t)$. The information about the angular velocity is requested by the routine **main.f90**.

3. **viscous.f90** (MODULE): In this file, the shear viscosity function $\eta = \eta(\dot{\gamma}, t, \dots)$ is defined and calculated. This information is requested by **update.f90**.

4. **write2f.f90** (MODULE): This file takes care of writing all computed data into the different files. It is only the source **main.f90** that makes such request.

5. **shear.f90** (MODULE): This routine calculates the shear rate $\dot{\gamma}_\mathrm{c}$ from the computed velocity profile V_k(i, j) $\mathbf{i}_\theta$. It is the program **update.f90** that makes the request.

6. **update.f90** (MODULE): This file sets up the system of algebraic Equations 7.28 to 7.31 (Page 161). This file also contains the Thomas algorithm that is used in solving this system.

7. **main.f90** (PROGRAM): This is the center of the whole software, holding and passing information to and from the different subroutines. Some subroutines interact directly with each other without going through the channels defined by **main.f90** (this applies mostly for the subroutines in the files **update.f90**, **shear.f90** and **viscous.f90**). The geometry of the viscometer, including the bottom cone, is defined in this part of the software.

The user only interacts with two files in the above list, namely the **param.f90** and **viscous.f90**. As such these files will be explained in more detail in Appendix A. The above files can be compiled in the order shown in the above. The code has been tested on 3 different compilers, listed below.

1. **NAGWare FTN90, Version 2.191**

2. **Microsoft Fortran Powerstation 4.0**

3. **Compaq Visual Fortran 6.5 and 6.6** (the previous Digital Visual Fortran)

All of the above compile and link the source codes without incidence. However while executing the program, then in some cases an abnormal termination occurs when using binaries from the **NAGWare** compiler. Of the three, the **Compaq Visual Fortran** is the most recommended one. Not only does it compile and link without incidence, but the user has also a better opportunity to control in what way the binaries are generated. Optimizing this compiler for speed, generates binaries which executes considerable faster than either of the two other compilers mentioned. The grid resolution ($\Delta r \wedge \Delta z$) used, is in accordance to the CPU resources available to this project.

8.3 Solution Geometry & Boundary Conditions

8.3.1 ConTec BML Viscometer 3

Figure 8.1 demonstrates the solution geometry and boundary conditions used when simulating the ConTec **BML** Viscometer 3. This array is 146×306 in size and the spacing of grid points in r and z-direction are uniform and equal: $\Delta r = \Delta z = 1.0\,\text{mm}$. This greatly simplifies the programming of the software, saves storage space and usually results in greater accuracy in the numerical calculation [2]. In the algorithm, the viscometer is rotated by 90° clockwise to simplify array addressing. This is apparent when comparing Figure 8.1 with the right illustration of Figure 8.15 (Page 197).

Referring to Figure 8.1: In the **i-direction**, the terms NX1 and NX2 designates the coordinates of the inner and outer cylinder and are given by 101 and 146, respectively. With these numbers, the geometry of the viscometer can be read directly from the figure. The inner radius is $R_\text{i} = (\text{NX1} - 1)\Delta r = (101 - 1)1.0\,\text{mm} = 10.0\,\text{cm}$ and likewise the outer radius is $R_\text{o} = (\text{NX2} - 1)\Delta r = (146 - 1)1.0\,\text{mm} = 14.5\,\text{cm}$.

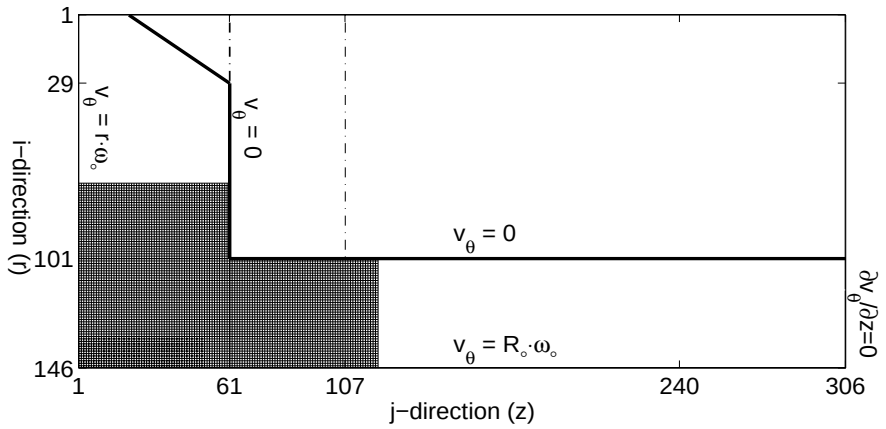


Figure 8.1: Geometry and boundary conditions for the ConTec **BML** Viscometer 3. This information is put into the arrays (146×306) VELOCITY_k, VELOCITY_kp12 and VELOCITY_kp1 of the source code **main.f90**. The lower left corner demonstrates the grid resolution used. For the domain of calculation, there are about 19500 grid points.

In the **j-direction**, the term NY1 designates the start location of the inner cylinder and NY2 designates the total height of the outer cylinder. These values are given by NY1=61 and NY2=306. The total height of the inner cylinder can now be calculated: $\text{H2} = (\text{NY2} - \text{NY1})\Delta z = (306 - 61)1.0\,\text{mm} = 24.5\,\text{cm}$. The term $h \equiv \text{H3} = 19.9\,\text{cm}$ designates the height where torque is measured and it starts at the **j-grid** point $\text{NY2mH3} = \text{NY2} - \text{H3}/\Delta z = 306 - 19.9\,\text{cm}/1\,\text{mm} = 107$ and ends at NY2=306. All the above number variables are shown in the source codes **main.f90** and **param.f90**.

Along the boundary (i,j) = (1:NX2,1)=(1:146,1) is the Dirichlet boundary condition: $v_\theta = r\,\omega_\text{o}(t)$, and at (i,j) = (NX2,1:NY2) = (146,1:306) the same type of boundary condition applies $v_\theta = R_\text{o}\,\omega_\text{o}(t)$. The open boundary (i.e. the interface between air and concrete) is located at (i,j) = (NX1+1:NX2-1,NY2) = (102:145,306) and there the Neumann boundary condition applies $\partial v_\theta/\partial z = 0$, and its justification will be discussed shortly. Everywhere else the boundary condition $v_\theta = 0$ is valid. All

the Dirichlet boundary conditions are set in accordance with the *no-slip condition*[1].

The combination of high viscosity of the fresh concrete and low rotational frequency of the outer cylinder $f_{\max} = 0.5\,\text{rps}$, results in that little or no inclination of the open boundary is observed, when making a rheological measurement[2]. Therefore, the normal vector of this open (or surface) boundary (i.e. at (i,j)=(NX1+1:NX2-1,NY2)) is set to be $\mathbf{n} \cong \mathbf{i}_z$. Since practically no shearing is applied from the atmospheric air onto the concrete at this boundary, then according to Cauchy's stress principle [72], applying the vector $\mathbf{n} = \mathbf{i}_z$ on Equation 7.4, must result in zero: $\mathbf{i}_z \cdot \mathbf{T} = 0$. This means that the condition $\partial v_\theta / \partial z = 0$ becomes valid[3] at the open boundary.

8.3.2 ConTec Viscometer 4

Figure 8.2 demonstrates the solution geometry and boundary conditions used when simulating the ConTec Viscometer 4. This array is 203×281 in size and the spacing of grid points in r and z-direction are uniform and equal: $\Delta r = \Delta z = 0.5\,\text{mm}$.

Referring to Figure 8.2, then in the **i-direction**, the terms NX1 and NX2 designate the coordinates of the inner and outer cylinder and are given by 171 and 203, respectively. Hence, the inner radius is $R_\text{i} = (\text{NX1}-1)\Delta r = (171-1)0.5\,\text{mm} = 8.5\,\text{cm}$ and the outer radius is $R_\text{o} = (\text{NX2}-1)\Delta r = (203-1)0.5\,\text{mm} = 10.1\,\text{cm}$.

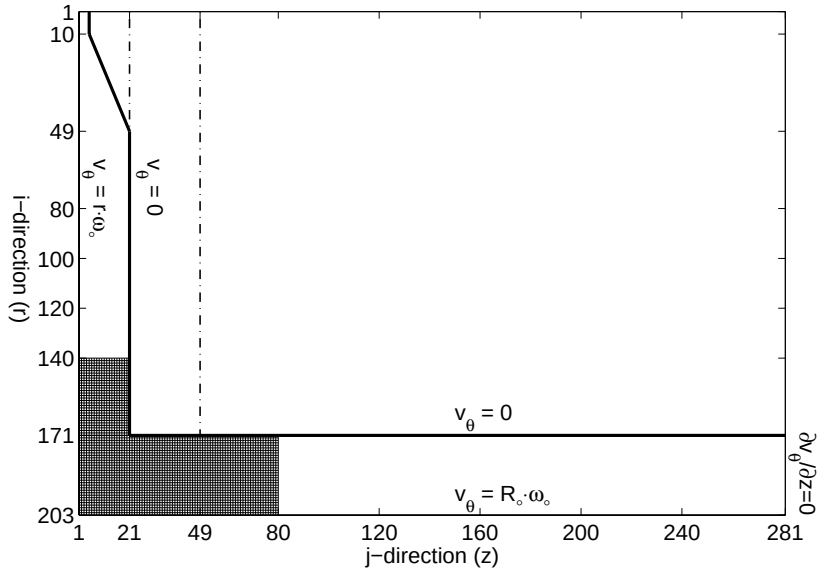


Figure 8.2: Solution geometry and boundary conditions for the viscometer. This information is put into the arrays (203×281) VELOCITY_k, VELOCITY_kp12 and VELOCITY_kp1 in the routine **main.f90**. The lower left corner demonstrates the grid resolution used. For the domain of calculation, there are about 11000 grid points.

[1] The velocity of a fluid at the solid surface is equal to the velocity of the surface. I.e. the fluid sticks to the surface and does not slip relative to it [36]. See also Footnote 9, Page 58.

[2] It is self evident that fluid material with $\eta \to \infty$ (i.e. a rigid body) gives no inclination.

[3] Another way in justifying this condition at the open boundary, is simply postulating that the velocity is not supposed to change with z in the top region of the viscometer. Since it is the shearing ($\mathbf{i}_z \cdot \mathbf{T} = T_{z\theta}\,\mathbf{i}_\theta$) from the bottom plate (at (i,j)=(1:NX2,1)), that is responsible for z-dependence in the velocity function v_θ, it is only natural that this dependency should decrease with increasing distance from this plate.

In the **j-direction** (i.e. in z-direction), the term NY1 designates the start location of the inner cylinder and NY2 designates the total height of the outer cylinder. These values are given by NY1 = 21 and NY2 = 281. The total height of the inner cylinder can now be calculated: $\mathrm{H2} = (\mathrm{NY2} - \mathrm{NY1})\Delta z = (281 - 21)0.5\,\mathrm{mm} = 13.0\,\mathrm{cm}$. As before, the term $h \equiv \mathrm{H3} = 11.6\,\mathrm{cm}$ designates the height where torque is measured and it starts at the **j-grid** point $\mathrm{NY2mH3} = \mathrm{NY2} - \mathrm{H3}/\Delta z = 281 - 11.6\,\mathrm{cm}/0.5\,\mathrm{mm} = 281 - 232 = 49$ and ends at NY2 = 281. All the coordinates related to the bottom cones are defined in **main.f90**. All other variables shown here, are defined in **param.f90**.

Along the boundary (i,j)=(1:NX2,1)=(1:203,1) the Dirichlet boundary condition $v_\theta = r\,\omega_\mathrm{o}(t)$ is valid. At (i,j)=(NX2,1:NY2)=(203,1:281) the same type of boundary condition applies, namely $v_\theta = R_\mathrm{o}\,\omega_\mathrm{o}(t)$.

The open boundary (i.e. the interface between air and mortar/cement paste) is located at (i,j)=(NX1+1:NX2-1,NY2)=(172:202,281). At this boundary, the Neumann boundary condition $\partial v_\theta/\partial z = 0$ applies, and its justification was discussed in Section 8.3.1. Everywhere else the condition $v_\theta = 0$ is valid.

8.4 Convergence and Stability

There is no guarantee that the **exact solution**[4] $v^\mathrm{k}_{\mathrm{i,j}}$ of the system of algebraic Equations 7.28 to 7.31 (Page 161) is close to the (unknown) **correct solution** $v_\theta(r,z,t)$ of the partial differential Equation 7.58 (Page 167). When the exact solution $v^\mathrm{k}_{\mathrm{i,j}}$ is approaching the correct solution $v_\theta(r,z,t)$ of the PDE as Δt, Δr and Δz are approaching zero, the exact solution is said to be **converging** [33, 66].

Note, in all the source codes presented in this thesis, the term "CONVERGENCE" is used in a slightly different context. In this context the meaning is that V_k(i,j) is approaching (or converging) towards the exact solution $v^\mathrm{k}_{\mathrm{i,j}}$ with increasing (pseudo) time iteration k, and as such is actually more related to stability.

Convergence is not to be confused with the **stability** of the **computed solution** V_k(i, j). When the computed solution V_k(i,j) approaches the exact solution $v^\mathrm{k}_{\mathrm{i,j}}$, for the given Δt, Δr and Δz, stability is achieved in the numerical calculations [33, 66]. More precisely, stability is the tendency for any spontaneous numerical perturbations $\xi^\mathrm{k}_{\mathrm{i,j}}$ in the computed solution $\mathrm{V\_k(i,j)} = v^\mathrm{k}_{\mathrm{i,j}} - \xi^\mathrm{k}_{\mathrm{i,j}}$ to decay. These perturbations consist mainly of the **round-off error** [2] and is not to be confused with the **discretization error** $e^\mathrm{k}_{\mathrm{i,j}} = v_\theta(r,z,t) - v^\mathrm{k}_{\mathrm{i,j}}$, which is due to incorrect logic of the discretization method used, as explained in Section 7.3.2 (Page 158).

8.4.1 Apparent Stability

The round-off error $\xi^\mathrm{k}_{\mathrm{i,j}} = v^\mathrm{k}_{\mathrm{i,j}} - \mathrm{V\_k(i,j)}$ exists because the computer cannot give answers to an infinite number of decimal places. Every calculation that is made, is carried out to a finite number of significant figures, which introduces a round-off error at each step of the computation [2, 33]. Hence, the computational solution of the system of algebraic Equations 7.28 to 7.31, is V_k(i, j) and not $v^\mathrm{k}_{\mathrm{i,j}}$. From the above, it is apparent that the relationship between the correct solution of the PDE, namely $v_\theta(r,z,t)$ and its computed counterpart V_k(i, j) is given by $v_\theta(r,z,t) = \mathrm{V\_k(i,j)} + [e^\mathrm{k}_{\mathrm{i,j}} + \xi^\mathrm{k}_{\mathrm{i,j}}]$. The computed shear rate $\dot\gamma_\mathrm{c}$, shear viscosity η_c, and the von Mises shear stress τ_c, are all calculated from V_k(i,j) using the DOUBLE PRECISION

[4] See Footnote 1 on Page 159.

declarations. In fact, all declarations in the different source codes, consist of DOUBLE PRECISION rather than REAL, to reduce the error $\xi^{k}_{i,j}$ in the computed velocity profile V_k(i,j). Also, a small time- and spatial steps Δt, Δr and Δz are used (usually with the constraint that $\Delta t/\Delta r^2 \leq 1\,\mathrm{s/m^2}$ and $\Delta t/\Delta z^2 \leq 1\,\mathrm{s/m^2}$) to reduce the discretization error $e^{k}_{i,j}$ (see discussion in Section 7.3.2).

In estimating the round-off error $\xi^{k}_{i,j}$ (or in other words, when investigating stability), the von Neumann- or matrix method is often used on a linearized version of the governing equation. The equation has to be linearized because these methods are strictly not applicable to non-linear equations [33]. In taking such approach here, the regularization parameter δ (Page 169) is quickly filtered out in the process. Experience from Section 8.5, gives that this parameter has a very large influence on stability and hence, such a *linearized* approach is useless at the best. When applying the Bingham fluid, experience indicates that stability is mainly function of δ, $\Delta t/\Delta r^2$, $\Delta t/\Delta z^2$ and τ_o/μ. When applying a viscoplastic-thixotropic fluid (i.e. thixotropic fluid with a yield value), additional viscosity parameters become also involved.

Since traditional stability analysis is not possible here, the root mean square of the difference between the two solutions V_k(i,j) and V_kp1(i,j) is monitored as a function of increasing (pseudo) time iteration k. This value serves as an indication to what degree stability exists in the calculation. This value is designated with the term RMS and is defined by Equation 7.76 (Page 173): $\mathrm{RMS} \equiv \|\mathrm{V\_k(i,j)} - \mathrm{V\_kp1(i,j)}\|_{\mathbf{rms}}$. In this work, the phrase **apparent stability** means that the RMS value decreases to a certain equilibrium value with increasing (pseudo) time iteration k.

Steady State Case: Output Files to Monitor Apparent Stability

The RMS value is written into a file called **log.dat**. The first column of this file consists of the iteration number k and the second column consists of the RMS value. The user can choose how often this file is written into, through the term k_OUTPUT_rms. Investigating stability by monitoring the RMS value, is only possible when consider the steady state case (i.e. time independence). Figure 8.3 demonstrates a typical RMS evolution with (pseudo) time. The number of (pseudo) time iterations shown is 10^6, i.e. $\Delta t \cdot 10^6 = 0.1\,\mathrm{s}$. The first RMS entry written into the file **log.dat** is here at k = 100 (since k_OUTPUT_rms = 100) and its value is $0.112 \cdot 10^{-2}$. This value and others that follows, are not shown in the figure to enhance the details. The minimum RMS value in the file is at $0.665 \cdot 10^{-11}$.

The oscillations of the RMS values shown in Figure 8.3 could indicate that a small error (probably closely related to the round-off error $\xi^{k}_{i,j}$) is propagating back and forth in the solution grid VELOCITY_k (see Figure 8.1). The origin of this error seem to be due to a very large values of shear viscosity $\eta_c = \mu + \tau_o/(\dot{\gamma}_c + \delta)$ produced when both the shear rate $\dot{\gamma}_c$ and the regularization parameter δ are very small. This can be concluded since numerical experiments give that when a larger regularization parameter δ is used, the oscillations reduce in amplitude. The same effect is produced when increasing the rotational frequency f_o, since it leads to a larger shear rate $\dot{\gamma}_c$. As shown in the small left illustration of Figure 8.3, the RMS value generally consists of different waves with distinct amplitudes and frequencies. Since k_OUTPUT_rms is equal to 100 and not equal to 1, the true frequency and phase variance of the RMS value are not shown. Using k_OUTPUT_rms = 1 for the given Δr and Δz, the frequency of the RMS value seems to increase with decreasing (pseudo) time step Δt. When Δr, Δz and Δt are all changing, the relationship between the frequency and the terms $\Delta t/\Delta r^2 \wedge \Delta t/\Delta z^2$ becomes somewhat unclear.

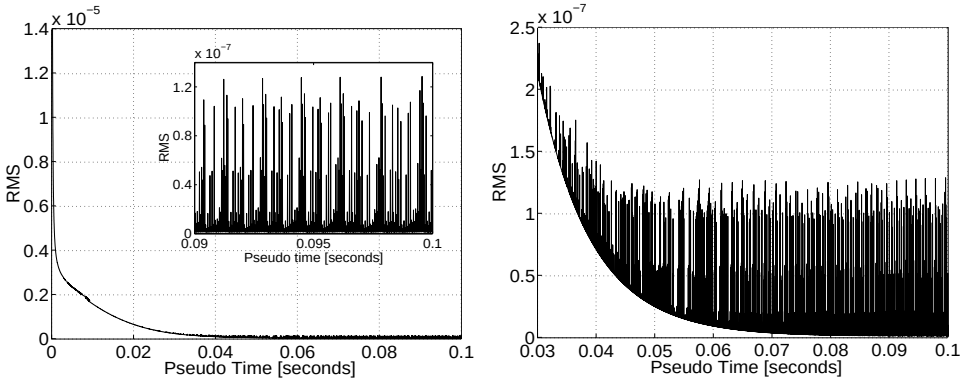


Figure 8.3: Registered RMS values extracted from the file **log.dat**. The physical parameters used are $\tau_o = 170\,\mathrm{Pa}$, $\mu = 35\,\mathrm{Pa}\cdot\mathrm{s}$ and $f_o = 0.1\,\mathrm{rps}$. The regularization parameter used, is $\delta = 10^{-3}\,\mathrm{s}^{-1}$. Both $\Delta t/\Delta r^2$ and $\Delta t/\Delta z^2$ are at $0.1\,\mathrm{s/m}^2$. The number of (pseudo) time iterations shown is 10^6.

To summarize, numerical experiments seem to indicate that the amplitude of the RMS value is reduced with increasing δ and f_o and with decreasing τ_o (or possibly rather τ_o/μ), $\Delta t/\Delta r^2$ and $\Delta t/\Delta z^2$.

As is shown with Figure 8.4, when the amplitudes of the RMS value become sufficiently large, an error becomes also visually apparent in the computed shear rate profile $\dot{\gamma}_c$. This is evident when a wave like phenomenon starts to appear in some of the solution region. Therefore, when doing a simulation, it is important to examine these oscillations and make sure that the RMS amplitudes are on a minimum level, and that the RMS evolution is approaching to a kind of equilibrium value as shown with the left illustration of Figure 8.3. Such a result can be enforced either by increasing the regularization parameter δ (which results in a worse imitation of the plastic- and the solid state) and/or by decreasing the time step Δt (which results in a longer calculation time). Increasing grid spacing Δr and Δz, will also result in greater stability. But as demonstrated in Sections 8.5 and 8.6, in doing so, a worse transition between the viscoplastic state (Ω_p) and solid state (Ω_e) is produced. The example in Figure 8.4 is taken from the BTRHEOM viscometer, presented Section 10.3.4. The physical parameters used are $\tau_o = 200\,\mathrm{Pa}$, $\mu = 20\,\mathrm{Pa}\cdot\mathrm{s}$ and $\omega_t = 3\,\mathrm{rad/s}$. The regularization parameter is at $\delta = 10^{-3}\,\mathrm{s}^{-1}$. Both $\Delta t/\Delta r^2$ and $\Delta t/\Delta z^2$ are set equal to $100\,\mathrm{s/m}^2$. This figure demonstrates that although the velocity profile is apparently smooth, it does not mean that there are no oscillations or errors in the calculations. Both the RMS value and the computed shear rate $\dot{\gamma}_c$ indicate strongly that some large errors are influencing the calculations. One approach to produce a better result, is reducing the existing time step $\Delta t = 10^{-4}\,\mathrm{s}$. Reducing it by an order of magnitude results in smaller wave amplitudes in the computed shear rate $\dot{\gamma}_c$.

Transient Case: Output Files to Monitor Apparent Stability

For the transient case (i.e. the time dependent calculations), the user has to check stability by investigating the numerical solution visually. In the case of ConTec viscometers, the software writes into 4 files for this purpose. These files are **vel_upper.dat**, **vel_corner.dat**, **ROS_upper.dat**, and **ROS_corner.dat** and are cumulative in nature, gathering velocity and shear rate information at chosen region inside the solution

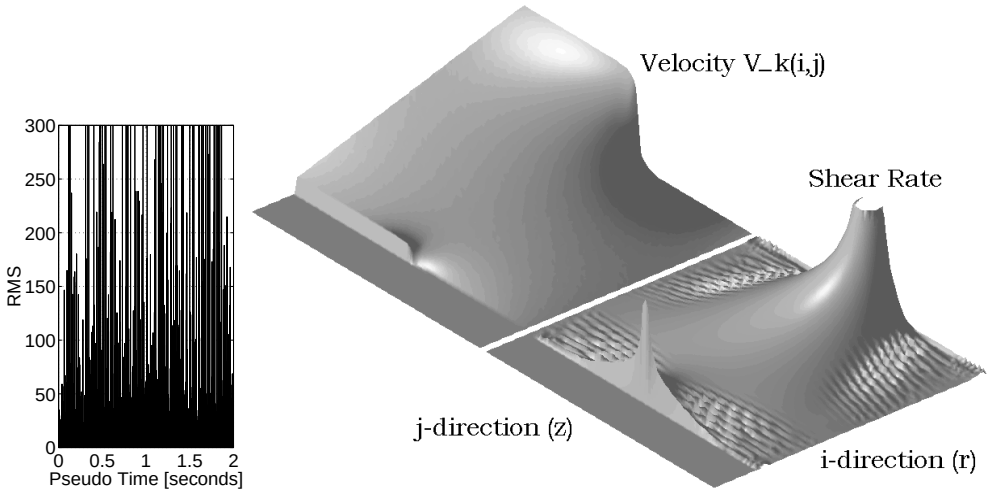


Figure 8.4: To the left: Registered RMS values extracted from the file **log.dat**. To the right: Normalized velocity profile V_k(i,j) and shear rate $\dot{\gamma}_c$, when solving for the BTRHEOM viscometer (see Section 10.3.4). The peak values shown, consist of $36\,\mathrm{cm/s}$ and $20\,\mathrm{s}^{-1}$.

domain, as a function of time $\mathrm{k}\cdot\Delta t$. The user can choose how often this file is written into, with the term dt_OUTPUT_torque. In this work, dt_OUTPUT_torque is usually set equal to 0.1, meaning that at every 0.1 calculated second, the file is written into. The file **vel_upper.dat** consists of the velocity profile V_k(i,j) at the top region, namely at (i,j) = (NX2-50:NX2,NY2-50:NY2), while the file **vel_corner.dat** consists of velocity profile at the bottom region (i,j) = (NX2-50:NX2,1:51). From these values, it is apparent that for every dt_OUTPUT_torque time, a segment of 51×51 array is added into the files. The size of this segment can be increased or decreased in the source code **write2f.f90** (see the subroutines WRITE2FILE_torque_ZERO and WRITE2FILE_torque). The files **ROS_upper.dat** and **ROS_corner.dat**, consist of the shear rate $\dot{\gamma}_c$ at the same locations as for the corresponding two above velocity files. As shown by Figure 8.4, these files are better suited to detect unwanted numerical oscillations.

8.4.2 Convergence

The PDE used in this work (i.e. Equation 7.6), cannot be solved directly by analytical means, and therefore is converted to an equivalent system of algebraic equations. It is a fundamental requirement of any numerical method that the exact solution $v^{\mathrm{k}}_{\mathrm{i,j}}$ of the system of algebraic equations can converge to the correct solution $v_\theta(r,z,t)$ of the PDE, as the grid parameters Δr, Δz and Δt, goes to zero [66]. With this, it is proven that the exact solution $v^{\mathrm{k}}_{\mathrm{i,j}}$ are very close to the (unknown) **correct solution** $v_\theta(r,z,t)$. However, for non-linear PDE like the one presented in this work, convergence is impossible to demonstrate theoretically [33]. Fortunately, it is possible to go through a kind of indirect route, where the **Lax Equivalence Theorem** is used as a necessary condition for convergence. It states "*Given a properly posed linear initial value problem and a finite difference approximation to it that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence*" [104] ([33, 77]). When working with non-linear PDE (as in this work), one cannot use the Lax equivalence theorem as a sufficient condition. But it can be interpreted as

providing the necessary condition for convergence [33].

If **apparent stability** (Section 8.4.1), is sufficient to verify stability, then stability applies for each numerical calculation in this work, unless otherwise stated. Also, in Section 7.6 (Page 163), consistency is proven in this thesis. With these two conditions, the Lax equivalence theorem can be applied, giving a necessary condition for convergence.

8.5 Numerical Convergence (Steady State Only)

The objective of demonstrating a numerical convergence is to reveal and investigate the capability of the numerical approach used in this work (see Chapter 7) in producing the correct solution of the PDE. Although the analysis here is applied only for the ConTec **BML** Viscometer 3, it is a good indication for the potential quality of the solution V_k(i,j) of all the other cases, shown in Chapters 9 and 10. This is because of how the same theory and same segments of source code is always used.

At the upper region of the ConTec viscometers, where z-independence exists ($\partial v_\theta/\partial z = 0$), it is possible to gain an analytical solution $v_\theta(r)$ when the test material consist of Bingham[5] fluid. The solution of such analytical case is shown in Chapter 3, namely with Equation 3.23 (Page 59). For this case, it is possible to demonstrate a **numerical convergence** by obtaining numerical solution on a successively refined grid and compute the solution error $RMS_{\mathbf{ac}}$, i.e. compute the difference between the exact analytical result $v_\theta(r)$ and the computed result V_k(i, j) (with j = constant) [33]. If numerical convergence exists, the solution error $RMS_{\mathbf{ac}}$ will reduce with reducing grid spacing [33]. This error is calculated according to Equation 8.1.

$$RMS_{\mathbf{ac}} \equiv \sqrt{\frac{1}{\mathrm{N}} \sum_{i \,\in\, \Omega} \left[\frac{v_\theta(r) - \mathrm{V\_k(i,j)}}{\frac{1}{2}(v_\theta(r) + \mathrm{V\_k(i,j)})} \right]^2} \quad \forall \quad r = [10.5{:}0.5{:}14.0]\,\mathrm{cm} \tag{8.1}$$

The $RMS_{\mathbf{ac}}$ value is calculated only at eight points, regardless of grid resolution used. These points starts at $r = 10.5\,\mathrm{cm}$ and with equal increments of 5 mm, ends at $r = 14.0\,\mathrm{cm}$. In doing this, the Dirichlet boundary points are not taken into account. Table 8.1 demonstrates $RMS_{\mathbf{ac}}$ results, when the rotational frequency is at $f_\mathrm{o} = 0.02\,\mathrm{rps}$, using different grid resolution Δr from 5 mm down to 0.5 mm.

In Section 8.4.1, the relationship between the correct solution of the PDE, namely $v_\theta(r, z, t)$ and its computed counterpart V_k(i, j) was given by $v_\theta(r, z, t) = \mathrm{V\_k(i,j)} + [e^{\mathrm{k}}_{\mathrm{i,j}} + \xi^{\mathrm{k}}_{\mathrm{i,j}}]$. Both the discretization error $e^{\mathrm{k}}_{\mathrm{i,j}}$ and the round-off error $\xi^{\mathrm{k}}_{\mathrm{i,j}}$ are inaccuracies generated by mathematical and computational means. However, since the computed solution V_k(i,j), with $\delta \neq 0$, is compared with analytical case v_θ using $\delta = 0$ (i.e. the ideal Bingham case), an additional error is presented here and is more of a physical nature. Hence, the difference between the desired solution $v_\theta(r, z, t)$ (with $\delta = 0$) and its computed counterpart V_k(i, j) (with $\delta \neq 0$), is rather given by $v_\theta(r, z, t) = \mathrm{V\_k(i,j)} + [e^{\mathrm{k}}_{\mathrm{i,j}} + \xi^{\mathrm{k}}_{\mathrm{i,j}} + d^{\mathrm{k}}_{\mathrm{i,j}}(\delta, f_\mathrm{o}, \dots)]$. The term $d^{\mathrm{k}}_{\mathrm{i,j}}(\delta, f_\mathrm{o}, \dots)$ will be known as the **regularization error** and has the following distinct characteristics: $d^{\mathrm{k}}_{\mathrm{i,j}}(0, f_\mathrm{o}, \dots) = 0 \wedge \partial d^{\mathrm{k}}_{\mathrm{i,j}}(\delta, f_\mathrm{o}, \dots)/\partial\delta > 0 \wedge \partial d^{\mathrm{k}}_{\mathrm{i,j}}(\delta, f_\mathrm{o}, \dots)/\partial f_\mathrm{o} < 0$. Using $\delta = 0$ in the numerical calculation will produce $d^{\mathrm{k}}_{\mathrm{i,j}} = 0$, however the round-off error $\xi^{\mathrm{k}}_{\mathrm{i,j}}$ will grow exponentially due to the extreme instabilities that will also be produced.

[5] This applies also when the test material consist of Newtonian fluid, but such a trivial case will not be considered here.

Table 8.1: Solution error when the rotational frequency is at $f_o = 0.02\,\mathrm{rps}$.

	$\Delta t/\Delta r^2$	5.0 mm	2.5 mm	1.0 mm	0.5 mm	δ
$RMS_{\mathbf{ac}}$	$0.1\,\mathrm{s\,m^{-2}}$	0.0090	0.0069	0.0044	0.0031	$10^{-3}\,\mathrm{s^{-1}}$
$RMS_{\mathbf{ac}}$	$1.0\,\mathrm{s\,m^{-2}}$	0.0669	0.0471	0.0295	0.0208	$10^{-2}\,\mathrm{s^{-1}}$
$RMS_{\mathbf{ac}}$	$1.0\,\mathrm{s\,m^{-2}}$	0.3251	0.2233	0.1388	0.0976	$10^{-1}\,\mathrm{s^{-1}}$
$\Delta\hat{T}$	$0.1\,\mathrm{s\,m^{-2}}$	0.61%	0.47%	0.42%	0.41%	$10^{-3}\,\mathrm{s^{-1}}$
$\Delta\hat{T}$	$1.0\,\mathrm{s\,m^{-2}}$	3.15%	2.95%	2.89%	2.89%	$10^{-2}\,\mathrm{s^{-1}}$
$\Delta\hat{T}$	$1.0\,\mathrm{s\,m^{-2}}$	19.05%	18.52%	18.36%	18.34%	$10^{-1}\,\mathrm{s^{-1}}$
$\tau_c(R_s)$	$0.1\,\mathrm{s\,m^{-2}}$	172.1 Pa	170.5 Pa	169.5 Pa	169.3 Pa	$10^{-3}\,\mathrm{s^{-1}}$
$\tau_c(R_s)$	$1.0\,\mathrm{s\,m^{-2}}$	167.2 Pa	165.7 Pa	165.2 Pa	165.1 Pa	$10^{-2}\,\mathrm{s^{-1}}$
$\tau_c(R_s)$	$1.0\,\mathrm{s\,m^{-2}}$	139.7 Pa	139.1 Pa	138.9 Pa	138.8 Pa	$10^{-1}\,\mathrm{s^{-1}}$

The analytical torque "$\hat{T}$ = constant" is calculated from Equation 3.24 and is compared with the numerical torque $\hat{T}_c(R_i)$ calculated from Equation 3.17 (Page 58), where $T_{r\theta}(R_i, z, t)$ is replaced with its corresponding numerical[6] counterpart $T^k_{i,j}$. The percentage difference is calculated according to Equation 8.2.

$$\Delta\hat{T} = 100 \cdot \frac{\hat{T} - \hat{T}_c(R_i)}{\hat{T}} \tag{8.2}$$

Always in this thesis, when calculating a partial derivative of any variable in the bulk, for example $\partial v_\theta/\partial r$, Equation 7.13 (Page 158) is used. As shown there, the truncation error consist of $O(\Delta r^2)$. When calculating the same partial derivative at the boundary $r = R_i$, Equation 7.52 (Page 166) is always used to maintain the same truncation error. This approach applies for all other boundary points as well, whenever a derivative needs to be calculated (see also Equations 7.53 and 7.54).

In investigating numerical convergence, the geometry of the ConTec **BML** Viscometer 3 is used. Also, the yield value and plastic viscosity used in this calculations consist of $\tau_o = 170\,\mathrm{Pa}$ and $\mu = 35\,\mathrm{Pa \cdot s}$. In this analysis, the bottom part of the viscometer is deleted in the source code, to decrease the calculation time. In doing so, the Neumann boundary condition $\partial v_\theta/\partial z = 0$ is applied at the new bottom part of the solution area. The implementation of this condition is done in the same manner as is done for the upper part, shown in Section 7.7 (Page 164). Using $f_o = 0.02\,\mathrm{rps}$, $\Delta r = \Delta z = 1\,\mathrm{mm}$, $\Delta t = 10^{-6}\,\mathrm{s}$, and $\delta = 10^{-2}\,\mathrm{s^{-1}}$, when comparing the solutions generated with the above approach with the solution generated when bottom area is included (with j = 245), demonstrates that no difference is encountered within five significant figures. This means that the same $RMS_{\mathbf{ac}}$ value is produced, namely $RMS_{\mathbf{ac}} = 0.0295$.

Rotational Frequency at $\mathbf{f_o = 0.02\,rps}$

By using Equation 3.32 (Page 66), the correct location of the boundary between the viscoplastic state (Ω_p) and the solid state (Ω_e) is calculated. With the rotational frequency of $f_o = 0.02\,\mathrm{rps}$, this location is calculated to be $R_s = 11.65\,\mathrm{cm}$ and is shown with the solid vertical lines in Figure 8.5.

[6]Since the condition $\partial v_\theta/\partial z = 0$ applies at the region in question and since $(\partial v_\theta/\partial r - v_\theta/r) \geq 0$, c.f. Section 3.3.3, it becomes apparent from Equations 7.5 and 7.7 (Page 156) that $\tau = T_{r\theta}$. Therefore the von Mises shear stress $\tau(r) = \eta(r)\,\dot{\gamma}(r)$ is used here, rather than the shear stress component $T_{r\theta}(r)$, when calculating the torque. This means that $\hat{T}_c = 2\,\pi\,r^2\,h\,T_{i,j} = 2\,\pi\,r^2\,h\,\tau_c$.

Table 8.1 demonstrates the $RMS_{\mathbf{ac}}$ and $\Delta\hat{T}$ values as a function of successively refined grid spacing, starting at $\Delta r = 5.0\,\mathrm{mm}$ and ending at $\Delta r = 0.5\,\mathrm{mm}$. Also, the last three rows of this table shows the computed von Mises shear stress τ_c (see Equation 7.5, Page 156) at $r = R_\mathrm{s}$. Since the yield value of the test material is $\tau_\mathrm{o} = 170\,\mathrm{Pa}$, then accordingly the correct von Mises shear stress at this location should also be $\tau(R_\mathrm{s}) = \tau_\mathrm{o} = 170\,\mathrm{Pa}$. The reason for this is evident by Equation 3.6 (Page 54): When $\tau = -\mathrm{II}^\mathrm{P}_\mathbf{S} = (\mathbf{T} : \mathbf{T})/2 = \tau_\mathrm{o}^2$, a transition from viscoplastic state to solid state (or visa versa) should occur. Because of the regularization parameter δ, the shear viscosity used in the simulations, does not correspond to a pure Bingham fluid (i.e. the $\delta = 0$ case) and hence a difference between the correct von Mises shear stress $\tau(R_\mathrm{s}) = \tau_\mathrm{o} = 170\,\mathrm{Pa}$ and the corresponding computed value $\tau_\mathrm{c}(R_\mathrm{s})$ is evident ($e^\mathrm{k}_{\mathrm{i,j}} + \xi^\mathrm{k}_{\mathrm{i,j}}$ contributes also to the difference). Although using a value large as $\delta = 10^{-2}\,\mathrm{s}^{-1}$, the difference between $\tau_\mathrm{c}(R_\mathrm{s})$ and $\tau(R_\mathrm{s})$ is only at 2.9% in the worst case, as can be calculated from the table. The difference in torque is only at 3.15% for the same case.

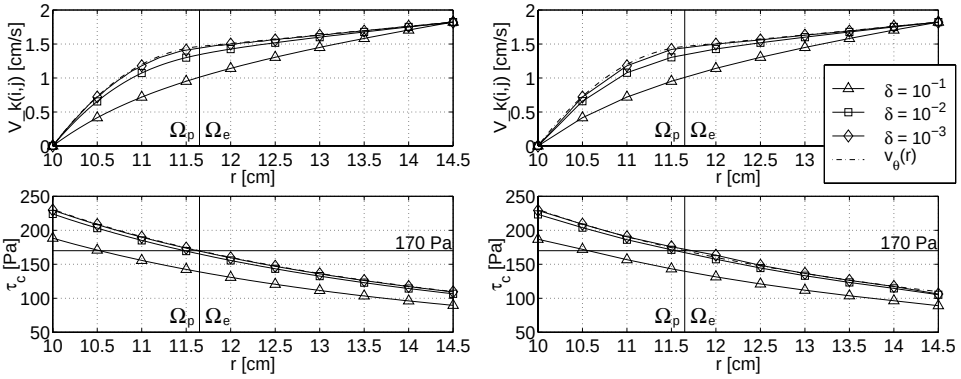


Figure 8.5: Top: Velocity profile V_k(i,j), when the grid spacing is $\Delta r = 0.5\,\mathrm{mm}$ (to the left) and $\Delta r = 5.0\,\mathrm{mm}$ (to the right). Bottom: The corresponding von Mises shear stress τ_c. The rotational frequency for all illustrations is $f_\mathrm{o} = 0.02\,\mathrm{rps}$. Dashed dotted lines represent the analytical results: $v_\theta(r)$ and $\tau(r)$.

Figure 8.5 demonstrates the velocity profile when using grid spacing of $\Delta r = 0.5\,\mathrm{mm}$ (top left figure) and when using $\Delta r = 5.0\,\mathrm{mm}$ (top right figure). As to be expected, both illustrations show how the computed velocity profile V_k(i,j) deviates from the correct velocity profile $v_\theta(r)$ as the regularization parameter δ is increased. Only a small deviation is evident, when using $\delta = 10^{-2}\,\mathrm{s}^{-1}$. Using $\delta = 10^{-3}\,\mathrm{s}^{-1}$, produces a computed solution V_k(i,j) which is further closer the analytical value $v_\theta(r)$, however at the cost of larger computational time. This is so, because using a smaller regularization parameter δ also means using a smaller (pseudo) time step Δt to maintain stability in the calculation.

Also shown in Figure 8.5, are the computed von Mises shear stress profile τ_c (solid lines) and its analytical counterpart τ (dashed dotted line). The analytical shear stress profile $\tau^2 = (\mathbf{T} : \mathbf{T})/2 = (\eta\,\dot{\gamma})^2$ is calculated with Equations 3.3 and 3.21 (Pages 53 and 58). For the reasons mentioned in Section 7.9 (Page 169), this calculation is only valid in the viscoplastic zone $r \in [R_\mathrm{i}, R_\mathrm{s}]$ (or $r \in \Omega_\mathrm{p}$). The analytical von Mises shear stress in the solid zone $r \in [R_\mathrm{s}, R_\mathrm{o}]$ (or $r \in \Omega_\mathrm{e}$) cannot be calculated directly, since the constitutive equation $\boldsymbol{\sigma}^\mathrm{E}$ and the displacement vector $\mathbf{u}$ in this region is unknown (Equation 3.1 on Page 52, may or may not be valid in this region).

However, with a known torque value $\hat{T}$ (from Equations 3.24 and 3.32, Pages 59 and 66), that applies both in the viscoplastic zone (Ω_p) and in the solid zone (Ω_e) (c.f. Equation 3.15, Page 58), it is possible to calculate the shear stress that applies in the solid zone, simply with $\hat{T}/(2\,\pi\,r^2\,h)$. This particular shear stress profile is also plotted in Figures 8.5 and 8.6 by the dashed dotted line.

The solid horizontal lines is the correct von Mises shear stress at the boundary between the viscoplastic state and the solid state: $\tau(R_\mathrm{s}) = \tau_\mathrm{o} = 170\,\mathrm{Pa}$. The correct condition exists when the computed value τ_c goes through the crossing of the vertical and the horizontal lines, as is the case when $\delta = 10^{-3}\,\mathrm{s}^{-1}$ and $\Delta r = 0.5\,\mathrm{mm}$.

It is interesting to note that although the $RMS_{\mathbf{ac}}$ and $\hat{T}$ values are decreasing with decreasing grid spacing Δr, for all δ values concern, such clear tendency is not produced for $\tau_\mathrm{c}(R_\mathrm{s})$. For $\delta = 10^{-3}\,\mathrm{s}^{-1}$, the value of $\tau_\mathrm{c}(R_\mathrm{s})$ is closest to 170 Pa when using $\Delta r = 2.5\,\mathrm{mm}$ and $\Delta r = 1\,\mathrm{mm}$. For $\delta = 10^{-2}\,\mathrm{s}^{-1}$ and $\delta = 10^{-1}\,\mathrm{s}^{-1}$, the best value is produced with the largest grid spacing, namely of $\Delta r = 5\,\mathrm{mm}$. At first consideration, this result is not to be expected. However, the reason for this peculiar tendency is as follows: First of all, the computed shear rate $\dot{\gamma}_\mathrm{c}$ is calculated to be greater when using a coarse grid spacing. This increase $\Delta\dot{\gamma}_\mathrm{c}^\mathrm{t} > 0$ is simply a consequence of the truncation error produced, as the information about the computed velocity V_k(i,j) in the very near vicinity of $r = R_\mathrm{s}$ is missing, when using the grid spacing of $\Delta r = 5\,\mathrm{mm}$ (see Equation 7.13, Page 158). Secondly, in estimating the shear rate at $r = R_\mathrm{s}$, a linear interpolation is always used between the two closest grid points (see Figure 8.11) which elevates the computed shear rate further by $\Delta\dot{\gamma}_\mathrm{c}^\mathrm{i} > 0$. Since $d\tau = [\mu + \tau_\mathrm{o}\,\delta/(\dot{\gamma}+\delta)^2]\,d\dot{\gamma} \approx \text{constant}\cdot d\dot{\gamma}$, the error $\Delta\dot{\gamma}_\mathrm{c} = \Delta\dot{\gamma}_\mathrm{c}^\mathrm{t} + \Delta\dot{\gamma}_\mathrm{c}^\mathrm{i} > 0$ results in an increase of $\Delta\tau_\mathrm{c} > 0$ in the computed (or estimated) von Mises shear stress $\tau_\mathrm{c}(R_\mathrm{s})$. To summarize, with increased regularization parameter δ, the computed shear stress τ_c decreases. However, with concomitant increase in grid spacing Δr, the shear stress value gains an addition of $\Delta\tau_\mathrm{c} > 0$ due to the truncation (and interpolation) error. Hence, using $\Delta r = 5\,\mathrm{mm}$ with either $\delta = 10^{-2}\,\mathrm{s}^{-1}$ or $\delta = 10^{-1}\,\mathrm{s}^{-1}$ produces the most correct result for the wrong reasons.

Rotational Frequency at $\mathbf{f_o = 0.1\,rps}$

Again, using Equation 3.32 (Page 66), it becomes apparent that plug is also occurring when the rotational frequency is at $f_\mathrm{o} = 0.1\,\mathrm{rps}$. The correct location of the boundary between the plastic- and solid state is now calculated to be $R_\mathrm{s} = 13.79\,\mathrm{cm}$. For this particular case, Table 8.2 shows the $RMS_{\mathbf{ac}}$ and $\Delta\hat{T}$ values as a function of successively refined grid spacing. Since the correct von Mises shear stress at $r = R_\mathrm{s}$, is not influenced by velocity of the test material, it has the same value here as for the previous case, namely $\tau(R_\mathrm{s}) = \tau_\mathrm{o} = 170\,\mathrm{Pa}$.

As mentioned earlier, because of the regularization parameter δ, the fluid used in the simulations is not of a pure Bingham fluid and hence a difference between the ideal von Mises shear stress $\tau(R_\mathrm{s}) = \tau_\mathrm{o} = 170\,\mathrm{Pa}$ and the corresponding computed value $\tau_\mathrm{c}(R_\mathrm{s})$ is evident ($e_{\mathrm{i,j}}^\mathrm{k} + \xi_{\mathrm{i,j}}^\mathrm{k}$ contributes also to the difference). This is shown in Table 8.2 and Figure 8.6. As shown for the $f_\mathrm{o} = 0.02\,\mathrm{rps}$-case (Table 8.1), non acceptable results are produced when the regularization parameter δ is as high as $10^{-1}\,\mathrm{s}^{-1}$. Increasing the rotational frequency up to $f_\mathrm{o} = 0.1\,\mathrm{rps}$ diminishes this error significantly. This is apparent when looking at the figure, where a smaller difference between the analytical result $v_\theta(r)$ and its computed counterpart V_k(i,j) is produced. The reason for this is as follows: With increasing rotational frequency f_o, the shear rate $\dot{\gamma}$ increases, which leads to $\eta = \mu + \tau_\mathrm{o}/(\dot{\gamma}+\delta) \approx \mu + \tau_\mathrm{o}/\dot{\gamma}$. In other words, as the

Table 8.2: Solution error when the rotational frequency is at $f_o = 0.1\,\mathrm{rps}$.

	$\Delta t/\Delta r^2$	5.0 mm	2.5 mm	1.0 mm	0.5 mm	δ
$RMS_{\mathbf{ac}}$	$0.1\,\mathrm{s\,m^{-2}}$	0.0025	0.0020	0.0013	0.0009	$10^{-3}\,\mathrm{s^{-1}}$
$RMS_{\mathbf{ac}}$	$1.0\,\mathrm{s\,m^{-2}}$	0.0172	0.0120	0.0075	0.0053	$10^{-2}\,\mathrm{s^{-1}}$
$RMS_{\mathbf{ac}}$	$1.0\,\mathrm{s\,m^{-2}}$	0.0822	0.0565	0.0352	0.0247	$10^{-1}\,\mathrm{s^{-1}}$
$\Delta\hat{T}$	$0.1\,\mathrm{s\,m^{-2}}$	0.40%	0.22%	0.16%	0.15%	$10^{-3}\,\mathrm{s^{-1}}$
$\Delta\hat{T}$	$1.0\,\mathrm{s\,m^{-2}}$	1.24%	1.05%	0.99%	0.98%	$10^{-2}\,\mathrm{s^{-1}}$
$\Delta\hat{T}$	$1.0\,\mathrm{s\,m^{-2}}$	6.10%	5.89%	5.83%	5.82%	$10^{-1}\,\mathrm{s^{-1}}$
$\tau_c(R_s)$	$0.1\,\mathrm{s\,m^{-2}}$	172.3 Pa	170.6 Pa	169.9 Pa	169.8 Pa	$10^{-3}\,\mathrm{s^{-1}}$
$\tau_c(R_s)$	$1.0\,\mathrm{s\,m^{-2}}$	169.8 Pa	168.7 Pa	168.4 Pa	168.4 Pa	$10^{-2}\,\mathrm{s^{-1}}$
$\tau_c(R_s)$	$1.0\,\mathrm{s\,m^{-2}}$	160.8 Pa	160.3 Pa	160.1 Pa	160.1 Pa	$10^{-1}\,\mathrm{s^{-1}}$

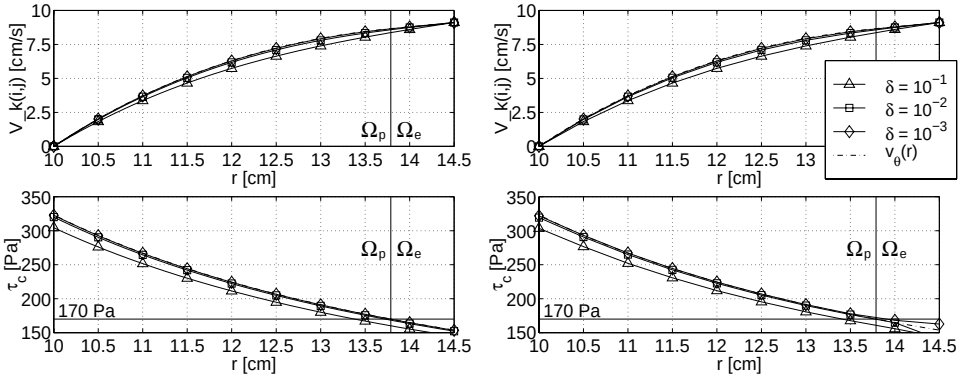


Figure 8.6: Top: Velocity profile V_k(i,j), when the grid spacing is $\Delta r = 0.5\,\mathrm{mm}$ (to the left) and $\Delta r = 5.0\,\mathrm{mm}$ (to the right). Bottom: The corresponding von Mises shear stress τ_c. The rotational frequency for all illustrations is $f_o = 0.1\,\mathrm{rps}$. Dashed dotted lines represent the analytical results: $v_\theta(r)$ and $\tau(r)$.

shear rate grows larger, the shear viscosity used in the numerical simulation becomes more closer to the ideal Bingham shear viscosity.

As before, the $RMS_{\mathbf{ac}}$ and $\hat{T}$ values are decreasing with decreasing grid spacing Δr, for all δ values concern. Such clear tendency is not produced for $\tau_c(R_s)$ and this is to be expected, c.f. the discussion for the $f_o = 0.02\,\mathrm{rps}$-case.

Rotational Frequency at $\mathbf{f_o = 0.25\,rps}$ and $\mathbf{f_o = 0.5\,rps}$

Increasing the rotational frequency further up to 0.25 rps, eliminates plug. This is apparent when using Equation 3.32, which gives $R_s > R_o$ for this particular case. Therefore the numerical von Mises shear stress $\tau_c(R_s)$ is not shown in Table 8.3. Since the $\Delta t/\Delta r^2$ value is kept at $1\,\mathrm{s/m^2}$ for all cases, its value is neither shown.

With $f_o = 0.25\,\mathrm{rps}$ and then with $f_o = 0.5\,\mathrm{rps}$, the effect from the regularization parameter becomes smaller and smaller. This is apparent from Table 8.3, where the largest difference between the correct torque $\hat{T}$ and the corresponding numerical counterpart $\hat{T}_c$ is only at 1.8%. The reason for this has been discussed previously and is related to that the shear viscosity used in the simulation becomes more equal to the ideal shear viscosity η of the Bingham fluid, as the shear rate $\dot{\gamma}$ increases. In other

Table 8.3: Solution error when the rotational frequency is at $f_o = 0.25 \wedge 0.5\,\mathrm{rps}$.

	f_o [rps]	5.0 mm	2.5 mm	1.0 mm	0.5 mm	δ
$RMS_{\mathbf{ac}}$	0.25	$4.01 \cdot 10^{-4}$	$1.39 \cdot 10^{-4}$	$0.65 \cdot 10^{-4}$	$0.43 \cdot 10^{-4}$	$10^{-3}\,\mathrm{s}^{-1}$
$RMS_{\mathbf{ac}}$	0.25	$16.0 \cdot 10^{-4}$	$9.91 \cdot 10^{-4}$	$5.95 \cdot 10^{-4}$	$4.15 \cdot 10^{-4}$	$10^{-2}\,\mathrm{s}^{-1}$
$RMS_{\mathbf{ac}}$	0.25	$126 \cdot 10^{-4}$	$85.0 \cdot 10^{-4}$	$53.0 \cdot 10^{-4}$	$37.0 \cdot 10^{-4}$	$10^{-1}\,\mathrm{s}^{-1}$
$\Delta\hat{T}$	0.25	0.29%	0.09%	0.03%	0.02%	$10^{-3}\,\mathrm{s}^{-1}$
$\Delta\hat{T}$	0.25	0.44%	0.24%	0.17%	0.17%	$10^{-2}\,\mathrm{s}^{-1}$
$\Delta\hat{T}$	0.25	1.80%	1.59%	1.53%	1.52%	$10^{-1}\,\mathrm{s}^{-1}$
$RMS_{\mathbf{ac}}$	0.50	$2.40 \cdot 10^{-4}$	$0.56 \cdot 10^{-4}$	$0.15 \cdot 10^{-4}$	$0.09 \cdot 10^{-4}$	$10^{-3}\,\mathrm{s}^{-1}$
$RMS_{\mathbf{ac}}$	0.50	$4.81 \cdot 10^{-4}$	$2.22 \cdot 10^{-4}$	$1.20 \cdot 10^{-4}$	$0.82 \cdot 10^{-4}$	$10^{-2}\,\mathrm{s}^{-1}$
$RMS_{\mathbf{ac}}$	0.50	$28.0 \cdot 10^{-4}$	$18.0 \cdot 10^{-4}$	$11.0 \cdot 10^{-4}$	$7.89 \cdot 10^{-4}$	$10^{-1}\,\mathrm{s}^{-1}$
$\Delta\hat{T}$	0.50	0.27%	0.08%	0.02%	0.01%	$10^{-3}\,\mathrm{s}^{-1}$
$\Delta\hat{T}$	0.50	0.32%	0.12%	0.06%	0.05%	$10^{-2}\,\mathrm{s}^{-1}$
$\Delta\hat{T}$	0.50	0.75%	0.55%	0.50%	0.48%	$10^{-1}\,\mathrm{s}^{-1}$

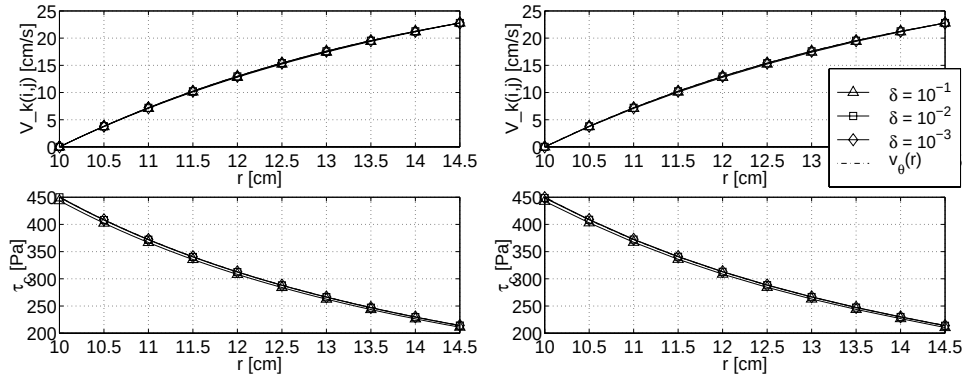


Figure 8.7: Top: Velocity profile V_k(i,j), when the grid spacing is $\Delta r = 0.5\,\mathrm{mm}$ (to the left) and $\Delta r = 5.0\,\mathrm{mm}$ (to the right). Bottom: The corresponding von Mises shear stress τ_c. The rotational frequency for all illustrations is $f_o = 0.25\,\mathrm{rps}$. Dashed dotted lines represent the analytical results: $v_\theta(r)$ and $\tau(r)$.

words, the shear rate $\dot{\gamma}$ overshadows the regularization parameter δ in the viscosity function η. Figure 8.7 shows the calculated velocity profile V_k(i,j) and the von Mises shear stress τ_c when $f_o = 0.25\,\mathrm{rps}$. A corresponding figure in the case of $f_o = 0.5\,\mathrm{rps}$ is shown in Figure 8.8. There, all computed lines V_k(i,j) coincide more or less with each other and with the correct solution $v_\theta(r)$.

Solution Error $\mathbf{RMS_{ac}}$

When looking at Table 8.1 to Table 8.3 it becomes clear that solution error $RMS_{\mathbf{ac}}$ becomes smaller with successively refined grid spacing Δr. This result is produced regardless of the magnitude of regularization parameter δ used. Figure 8.9 demonstrates the result from the tables when the rotational frequency is at $f_o = 0.02\,\mathrm{rps}$ (to the left) and at $f_o = 0.5\,\mathrm{rps}$ (to the right). There it becomes apparent how the solution error $RMS_{\mathbf{ac}}$ decreases with **1)** increased rotational frequency, **2)** with decreasing

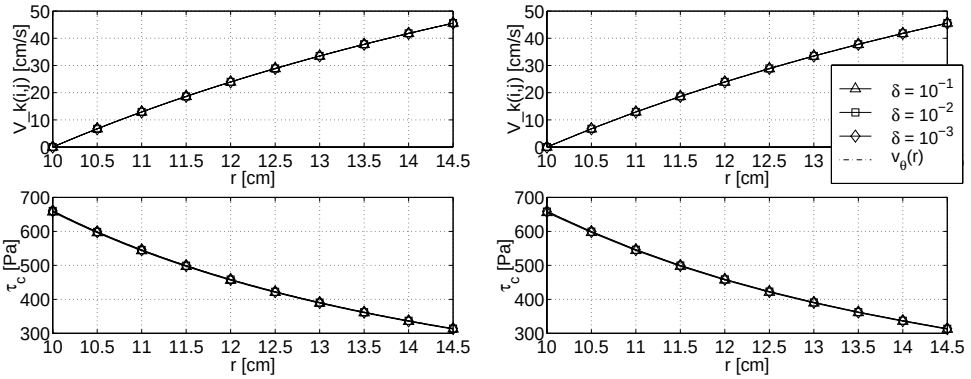


Figure 8.8: Top: Velocity profile V_k(i,j), when the grid spacing is $\Delta r = 0.5\,\mathrm{mm}$ (to the left) and $\Delta r = 5.0\,\mathrm{mm}$ (to the right). Bottom: The corresponding von Mises shear stress τ_c. The rotational frequency for all illustrations is $f_\mathrm{o} = 0.5\,\mathrm{rps}$. Dashed dotted lines presents the analytical results: $v_\theta(r)$ and $\tau(r)$.

regularization parameter δ and **3)** with successively refined grid spacing Δr.

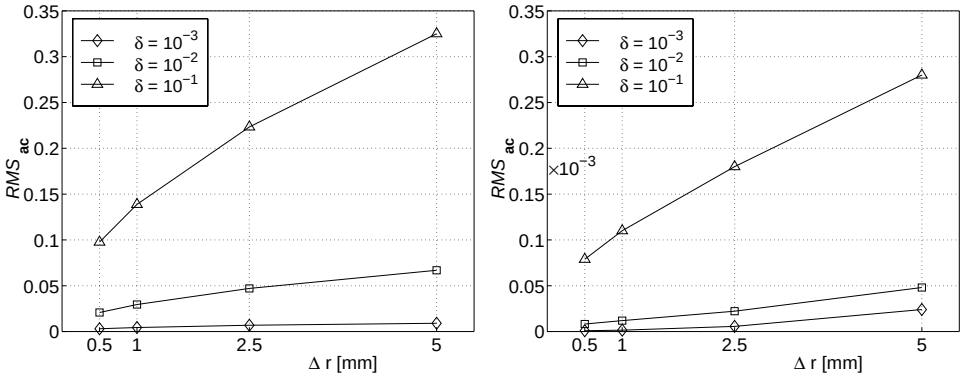


Figure 8.9: To the left: $RMS_{\mathbf{ac}}$ values at $f_\mathrm{o} = 0.02\,\mathrm{rps}$. To the right: $RMS_{\mathbf{ac}}$ values at $f_\mathrm{o} = 0.5\,\mathrm{rps}$. Note that in this latter figure, the scale of the ordinate is in 10^{-3} as shown in Table 8.3.

With the above, a numerical convergence is verified when using a Bingham fluid at the top region of the viscometer. Since exactly the same type of boundary condition and exactly the same programming routines are used in the bottom region of the whole solution domain (and also for other types of viscometric geometries in this work), it is reasonable to assume a numerical convergence applies also for this zone.

8.6 Imitation of the Viscoplastic and Solid State

In the viscoplastic zone $r \in [R_\mathrm{i}, R_\mathrm{s}]$ (or $r \in \Omega_\mathrm{p}$) and in the solid zone $r \in [R_\mathrm{s}, R_\mathrm{o}]$ (or $r \in \Omega_\mathrm{e}$) the question is how well does the software VISCOMETRIC-VISCOPLASTIC-FLOW, simulate these two physical conditions, relative to applied torque $\hat{T}$. This is the subject of current section. In investigating this topic, the same viscometric values apply as was used in Section 8.5, namely $\tau_\mathrm{o} = 170\,\mathrm{Pa}$ and $\mu = 35\,\mathrm{Pa}\cdot\mathrm{s}$. Like

before, the spacing between grid points in r- and z-direction is equal in this analysis $\Delta r = \Delta z$.

8.6.1 Imitation of the Solid State

In the plugged zone $r \in [R_\mathrm{s}, R_\mathrm{o}]$ (or $r \in \Omega_\mathrm{e}$), the question is how well does the software simulate the solid state. In Section 8.5, comparison of numerical solution V_k(i, j) with known analytical solution v_θ was made. The comparison was also made for the solid region $r \in [R_\mathrm{s}, R_\mathrm{o}]$, where V_k(i, j) is compared with $v_\theta = r\,\omega_\mathrm{o}$. With the low solution error $RMS_{\mathbf{ac}}$ shown in Figure 8.9, it is apparent that solid state is simulated in a quite satisfactory manner, when $\delta \leq 10^{-2}$. However, since a real viscometric measurement is based on logged torque, the quality of solid state might be best verified by demonstrating that the computed torque is equal to the analytical torque $\hat{T}_\mathrm{c} = \hat{T}$ and that it is a constant (i.e. $\hat{T}_\mathrm{c}$ = constant $\forall\ r \in [R_\mathrm{i}, R_\mathrm{o}]$) as shown in Equation 3.15 (Page 58). As proven there, the last condition must apply in both the plastic- and the solid domain.

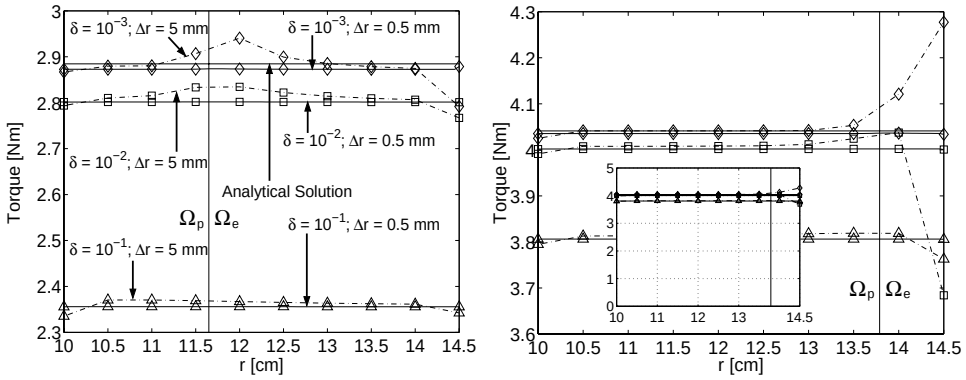


Figure 8.10: Torque $\hat{T}_\mathrm{c}$ as a function of the radius r, when $f_\mathrm{o} = 0.02\,\mathrm{rps}$ (to the left) and $f_\mathrm{o} = 0.1\,\mathrm{rps}$ (to the right). Same legends apply for both figures.

Figure 8.10 demonstrates the computed torque $\hat{T}_\mathrm{c}$ as a function of the radius r, when $f_\mathrm{o} = 0.02\,\mathrm{rps}$ (to the left) and $f_\mathrm{o} = 0.1\,\mathrm{rps}$ (to the right). The solid vertical lines demonstrate the correct location of the boundary between the plastic- and the solid state $r = R_\mathrm{s}$ calculated with Equation 3.32 (see also Figures 8.5 and 8.6). With $f_\mathrm{o} = 0.02\,\mathrm{rps}$, $\Delta r = 0.5\,\mathrm{mm}$ and $\delta = 10^{-3}\,\mathrm{s}^{-1}$, the computed torque $\hat{T}_\mathrm{c}$ is very close to the correct analytical solution $\hat{T}$ with only 0.4% difference. Using this smallest grid spacing ($\Delta r = 0.5\,\mathrm{mm}$) gives a torque profile which is constant $\forall\ r \in [R_\mathrm{i}, R_\mathrm{o}]$. This result in not produced when using the largest grid spacing of 5.0 mm. However, the former grid spacing is much more computational expensive since more grid points are present and also since a smaller time step Δt is required to maintain stability. With the smallest grid spacing, the computed torque $\hat{T}$ is a constant, regardless of regularization parameter δ used. When using the largest grid spacing, a "humpback" near the correct boundary between the viscoplastic state and the solid state $r = R_\mathrm{s}$ is produced for both $\delta = 10^{-3}\,\mathrm{s}^{-1}$ and $\delta = 10^{-2}\,\mathrm{s}^{-1}$. Also, at the outer cylinder $r = R_\mathrm{o}$, some additional deviations from the constant torque are produced. This type of deviation is also produced at the inner cylinder $r = R_\mathrm{i}$, but still at a much smaller magnitude.

When $f_o = 0.1\,\text{rps}$, the torque deviations at $r = R_s$ and at $r = R_o$ seem to be added together since $R_s \approx R_o$, which results in a larger error. This phenomenon does not appear when the smallest grid spacing is used, which emphasis its benefits.

Going back to $f_o = 0.02\,\text{rps}$, the analytical torque is equal to $\hat{T} = 2.885\,\text{Nm}$ and with $\delta = 10^{-3}\,\text{s}^{-1}$ and $\delta = 10^{-2}\,\text{s}^{-1}$, the difference between the analytical value $\hat{T}(R_s)$ and its computed counterpart $\hat{T}_c$, when using the smallest grid spacing, is 0.4% and 2.9%, respectively. However for $\delta = 10^{-1}\,\text{s}^{-1}$, the difference is large as 18.3%. Increasing the rotational frequency up to $f_o = 0.1\,\text{rps}$ gives smaller difference between $\hat{T}$ and $\hat{T}_c$. This is to be expected since with larger rotational frequency f_o, the shear rate $\dot{\gamma}$ increases and hence: $\eta = \mu + \tau_o/(\dot{\gamma} + \delta) \approx \mu + \tau_o/\dot{\gamma}$. For both illustrations of Figure 8.10, then with $\delta = 10^{-1}\,\text{s}^{-1}$ and $\Delta r = 5\,\text{mm}$, the torque seems to gain a constant value across the boundary $r = R_s$. This is factual since the increased δ results in a worse imitation of the solid state and hence the transition from the viscoplastic state to the solid state becomes much smoother. This is apparent with the much smoother shear rate profile $\dot{\gamma}_c$ at $r = R_s$ as shown in Figure 8.11.

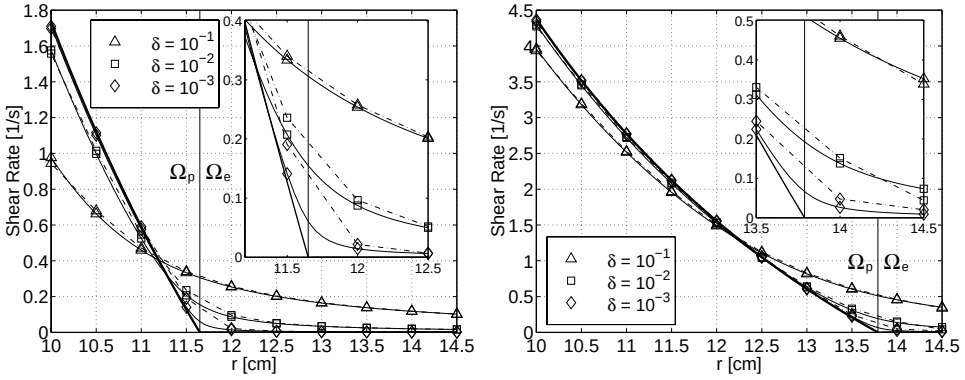


Figure 8.11: Shear rate $\dot{\gamma}_c$ as a function of the radius r, when $f_o = 0.02\,\text{rps}$ (to the left) and $f_o = 0.1\,\text{rps}$ (to the right). The thick solid line presents the analytical solution $\dot{\gamma}(r)$. Legends are shown in the left illustration of Figure 8.10.

Another approach is possible in validating the imitation of the solid state. As is shown with Equation 3.5 (Page 53), the shear rate must be zero $\dot{\gamma} = 0$ where pure solid state exists, namely in the domain of Ω_e. The computed shear rate $\dot{\gamma}_c$ is shown in Figure 8.11. With $\delta = 10^{-3}\,\text{s}^{-1}$, this condition is basically fulfilled. However, a small deviation exists, around the boundary between the viscoplastic state and the solid state. For $\delta = 10^{-2}\,\text{s}^{-1}$, the computed shear rate is not zero, but is close to it.

With $\delta = 10^{-1}\,\text{s}^{-1}$, the computed solution becomes unacceptably wrong. For most of the numerical calculations done in this thesis, the regularization parameter is set to be $\delta \leq 5 \cdot 10^{-3}\,\text{s}^{-1}$, which should produce a satisfactory imitation of the solid state and at the same time, keeping the calculation time within tolerable limits.

8.6.2 Imitation of the Viscoplastic State

Increasing the rotational frequency further up to 0.25 rps, eliminates plug. This is apparent when using Equation 3.32, which gives $R_s > R_o$ for this particular case. Therefore the whole domain $r \in [R_i, R_o]$ consists solely of a viscoplastic state. In this domain, the question is how well does the software simulate this condition. Figure 8.12

demonstrates the computed shear rate $\dot{\gamma}_c$ as a function of the radius r, when $f_o = 0.25\,\text{rps}$ (to the left) and $f_o = 0.5\,\text{rps}$ (to the right).

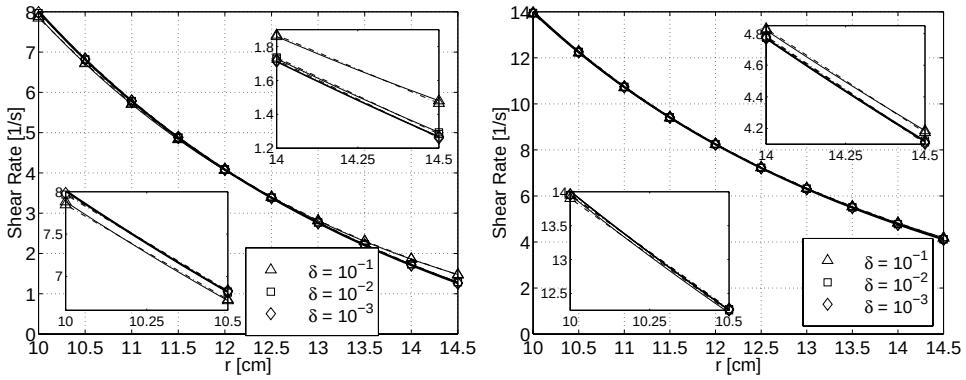


Figure 8.12: Shear rate $\dot{\gamma}_c$ as a function of the radius r, when $f_o = 0.25\,\text{rps}$ (to the left) and $f_o = 0.5\,\text{rps}$ (to the right). The thick solid line presents the analytical solution $\dot{\gamma}(r)$. Legends are shown in the left illustration of Figure 8.10.

As always, when calculating the shear rate at the boundaries $r = R_i$ and $r = R_o$, Equations 7.52 and 7.53 (Page 166) are applied as is shown with Equations 8.3 and 8.4 below. As mentioned in Section 8.5, then this is done to maintain a truncation error of $O(\Delta r^2)$. This results in the smooth shear rate profile at these boundaries as is shown in Figure 8.12.

$$\dot{\gamma}_c|_{R_i} = \frac{4\,v_{i+1,j} - v_{i+2,j} - 3\,v_{i,j}}{2\,\Delta r} - \frac{v_{i,j}}{R_i} \tag{8.3}$$

$$\dot{\gamma}_c|_{R_o} = \frac{-4\,v_{i-1,j} + v_{i-2,j} + 3\,v_{i,j}}{2\,\Delta r} - \frac{v_{i,j}}{R_o} \tag{8.4}$$

As shown in Figure 8.12, there is a very small difference between the computed shear rate $\dot{\gamma}_c$ and the analytical shear rate $\dot{\gamma}(r)$. This is to be expected, since with increasing rotational frequency f_o, the shear rate $\dot{\gamma}$ increases, and overshadows the regularization parameter δ in the viscosity function η. However, when using $\delta = 10^{-1}\,\text{s}^{-1}$, some deviation exists near and at the boundaries $r = R_i$ and $r = R_o$. This is apparent with the two small incorporated figures shown in each illustration:

Near and at the inner cylinder $r = R_i$, the computed shear rate $\dot{\gamma}_c$ is smaller than the (correct) analytical shear rate $\dot{\gamma}(r)$. Around the outer cylinder $r = R_o$, the opposite is true. This results applies regardless of spacing between grid points used $\Delta r = 0.5\,\text{mm}$ or $\Delta r = 5\,\text{mm}$. However, this deviation is reduced when the rotational frequency is increased up to $f_o = 0.5\,\text{rps}$. Using either $\delta = 10^{-2}\,\text{s}^{-1}$ or $\delta = 10^{-3}\,\text{s}^{-1}$, this type of deviation is not noticeable for either case of $f_o = 0.25\,\text{rps}$ or $f_o = 0.5\,\text{rps}$.

Looking back to Figure 8.11, the viscoplastic state is imitated in a quite satisfactory manner, when plug is occurring in the neighborhood. This applies when using either $\delta = 10^{-2}\,\text{s}^{-1}$ or $\delta = 10^{-3}\,\text{s}^{-1}$. Using larger regularization parameter $\delta = 10^{-1}\,\text{s}^{-1}$ leads to an unacceptable imitation of the viscoplastic state. For this last-mentioned case, the largest source of error is due to the incorrect boundary condition that apply at $r = R_s$, as a result of the smooth transition in velocity across this boundary (see Figures 8.5 and 8.6).

Figure 8.13 demonstrates the computed torque $\hat{T}_c$ as a function of radius r, when $f_o = 0.25\,\text{rps}$ (to the left) and $f_o = 0.5\,\text{rps}$ (to the right). As shown with the small incorporated figure shown in each illustration, the computed torque $\hat{T}_c$ is more or less equal to the analytical torque $\hat{T}$ for all cases. When applying the rotational frequency of $f_o = 0.25\,\text{rps}$, a small deviation exists for the $\delta = 10^{-1}\,\text{s}^{-1}$ case. This difference is further decreased when increasing the rotational frequency up to $f_o = 0.5\,\text{rps}$.

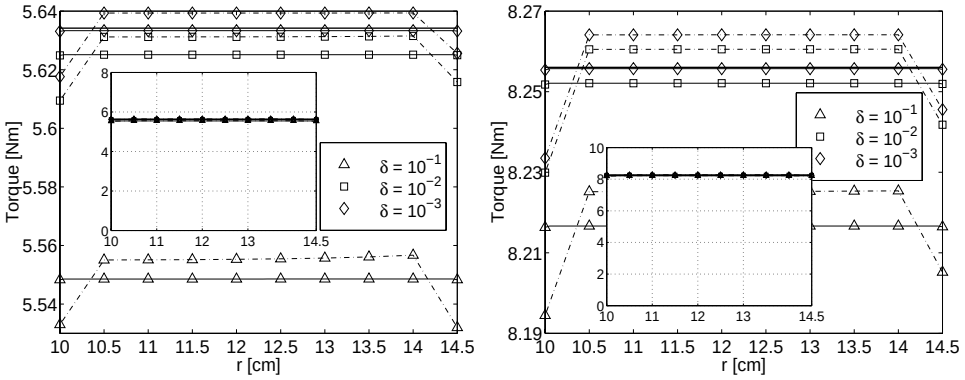


Figure 8.13: Torque $\hat{T}_c$ as a function of the radius r, when $f_o = 0.25\,\text{rps}$ (to the left) and $f_o = 0.5\,\text{rps}$ (to the right). Legends are shown in the left illustration of Figure 8.10.

When using the largest spacing between grid points $\Delta r = 5\,\text{mm}$, a drop in computed torque becomes factual at each boundary points $r = R_i$ and $r = R_o$. This is peculiar since the same truncation error of $O(\Delta r^2)$ applies when calculating the shear rate $\dot{\gamma}_c$ in the bulk as when calculating it at the boundary points[7]. Nevertheless, this drop consists of only $0.02/5.5 \approx 0.3\%$ in the worst case. When applying the smallest grid spacing, namely $\Delta r = 0.5\,\text{mm}$, this phenomenon is of no concern.

8.7 Some Numerical Results

The objective with this section is to examine how much z-dependency exists in the velocity function $v_\theta(r, z, t)$ in upper part of the viscometer. This is done by using the software VISCOMETRIC-VISCOPLASTIC-FLOW, shown in Appendix A. The inquiry has to do with justifying the assumption made in Section 3.3.1 (Page 56) when deriving the well-known Reiner-Riwlin equation. For Section 8.7.1, the viscometric values used consist of $\tau_o = 200\,\text{Pa}$, $\mu = 20\,\text{Pa}\cdot\text{s}$. However, for Section 8.7.2, they consist of $\tau_o = 60\,\text{Pa}$ and $\mu = 10\,\text{Pa}\cdot\text{s}$. In both cases, the regularization parameter is set equal to $\delta = 4\cdot 10^{-3}\,\text{s}^{-1}$.

8.7.1 ConTec BML Viscometer 3

Figures 8.14, 8.15 and 8.16 demonstrates three dimensional vector plot and isoplot of the computed velocity V_k(i,j) when the angular velocity ω_o is $1\,\text{rad/s}$. For the last-mentioned figure, the dashed dotted line demonstrates the (approximate) location of the boundary between the solid state Ω_e and the viscoplastic state Ω_p. As discussed

[7]It is the shear rate $\dot{\gamma}_c$ that the main dependent variable when calculating the torque $\hat{T}_c = 2\,\pi\,r^2\,h\,T_{i,j} = 2\,\pi\,r^2\,h\,\tau_c = 2\,\pi\,r^2\,h\,\eta(\dot{\gamma}_c)\,\dot{\gamma}_c$ (see also Footnote 6 on Page 186).

previously, the velocity profile of a pure solid state consist of $v_\theta = r\,\omega_o$ and is produced in the calculations, as noted with the even spaces between the isolines in the solid domain Ω_e.

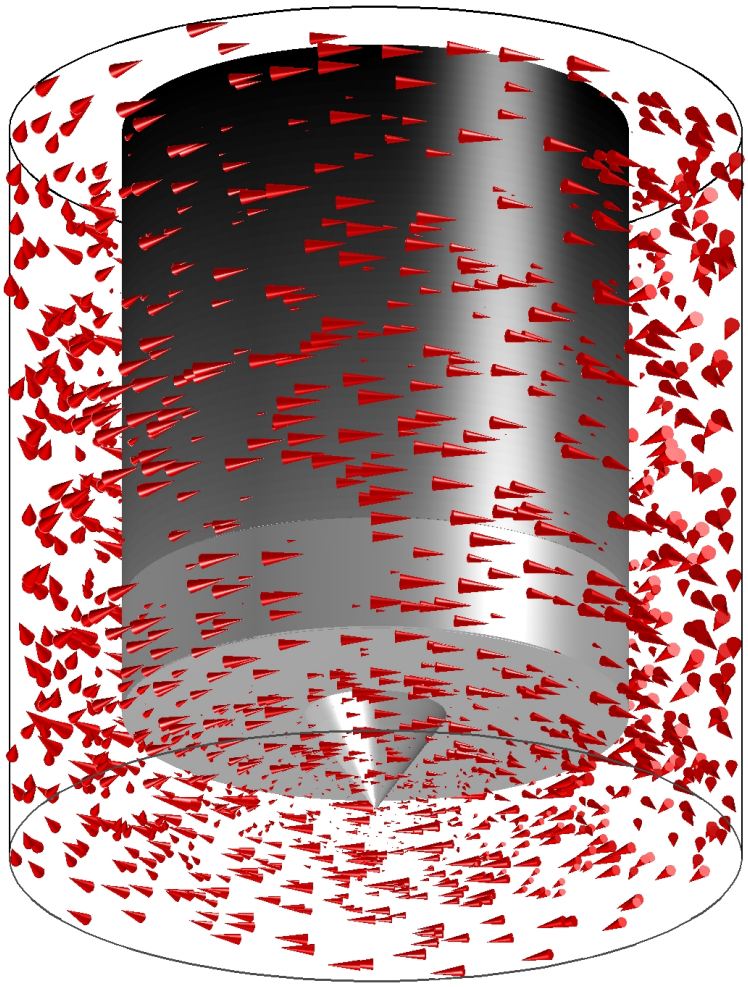


Figure 8.14: Three dimensional vector plot of velocity $\mathbf{v}$ (same condition as in Figure 8.16).

Before proceeding, there is one issue related to the simulation results presented in this section that needs to be discussed. The issue is apparent when observing a rheological experiment of concrete inside the ConTec **BML** Viscometer 3. This is perhaps also apparent when observing Figure 3.8 (Page 62), which demonstrates that the test material consist of coarse particle suspension with the largest aggregate size of $D_{max} = 16\,\text{mm}$. When monitoring the velocity of such a large aggregate particle, it is clear that its velocity $\mathbf{v}_I$ is of a random and spontaneous nature. This type of randomness is perhaps more present in the experiment because of the barrier restraint of the inner and outer cylinder (see discussion in Section 3.4.2). This random velocity is not to be confused with the smooth[8] velocity $\mathbf{v}$, which is shown for example in

[8]Discontinuities in the numerical velocity function V_k(i,j) can be produced as a result of the numerical error involved. Such discontinuities could mistakenly be understood as a correct solution of the random and spontaneous velocity type $\mathbf{v}_I$ of the individual solid particles. Any random and

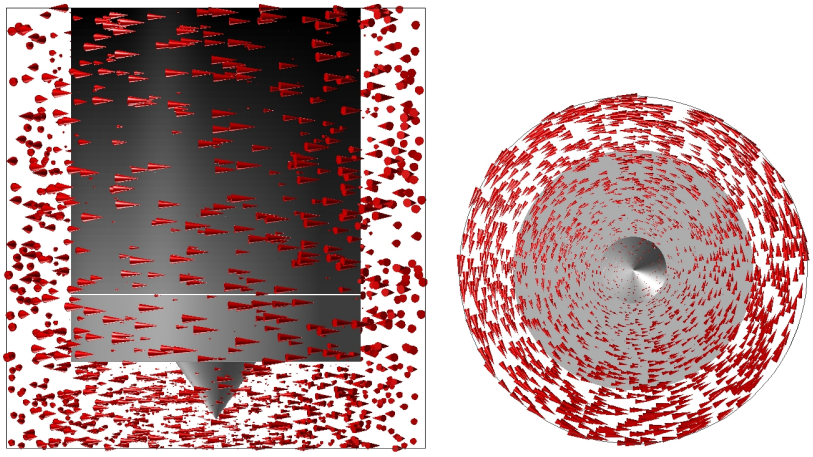


Figure 8.15: Vector plot of velocity $\mathbf{v}$ (same condition as in Figure 8.16).

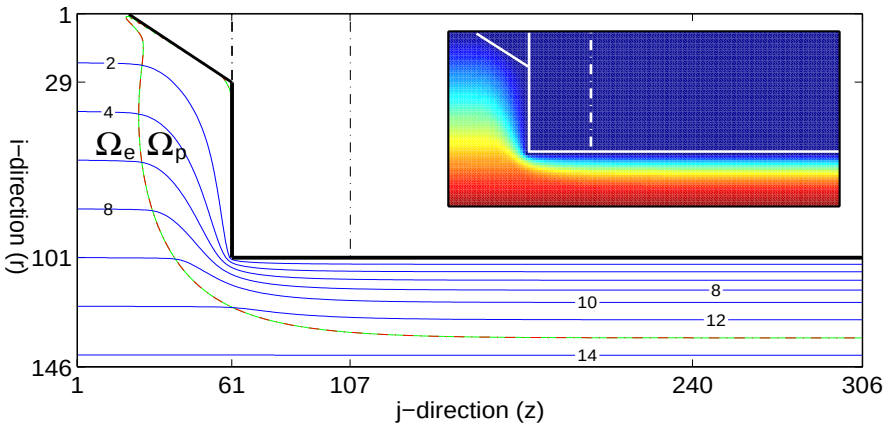


Figure 8.16: Velocity profile V_k(i,j) when $\omega_o = 1\,\mathrm{rad/s}$. The isolines start at $2\,\mathrm{cm/s}$ near the inner cylinder and with equal increments of $2\,\mathrm{cm/s}$, ends at $14\,\mathrm{cm/s}$ near the outer cylinder.

Figure 8.16 and calculated by Equation 2.18 (Page 16). As shown in Equation 2.6 (Page 13), this velocity $\mathbf{v}$ consist of the mass averaged velocity of all the solid particles composing the CP, for example shown in Figure 3.8. Hence, any random and spontaneous velocity contributions from the individual solid particles are summarized out in the averaging and only the relevant smooth motion of the continuum will remain. More precisely, when considering a coarse particle suspension like of the fresh concrete, the smooth velocity profile $\mathbf{v} = \mathbf{v}(r, z)$ like shown in Figure 8.16, does not represent the velocity for the individual solid particles (i.e. of the individual aggregate, cement grain or water molecule). This is because the latter type of motion is on a scale below what can be provided by Equation 2.18. The above discussion applies equally for water. One cannot use the Navier-Stokes Equation B.18 (Appendix B.6) to predict the random (i.e. thermal) motion of the individual water molecule, because

spontaneous velocity contribution cannot be calculated by Equation 2.18 (Page 16) because of the filter process made in the steps from Equation 2.2 to Equation 2.6 (Page 13).

such a motion is on a scale below what can be provided by this equation.

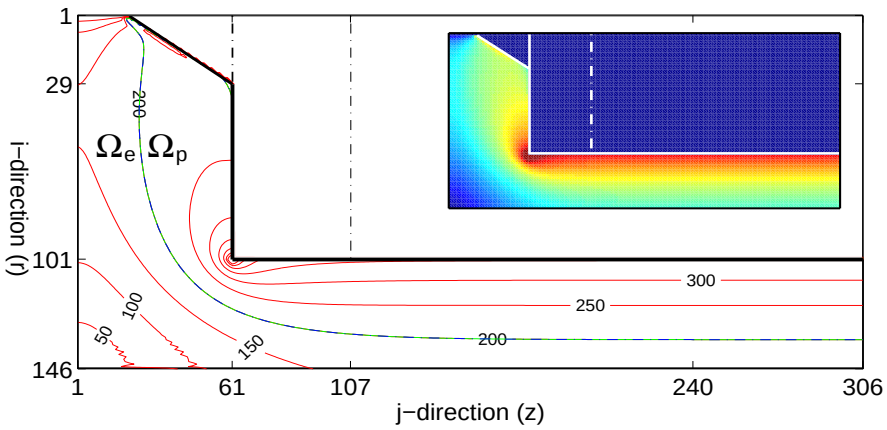


Figure 8.17: Shear stress τ_c when $\omega_o = 1\,\mathrm{rad/s}$. The isolines start at $50\,\mathrm{Pa}$ near the outer cylinder and with equal increments of $50\,\mathrm{Pa}$, ends at $500\,\mathrm{Pa}$ around the corner of inner cylinder.

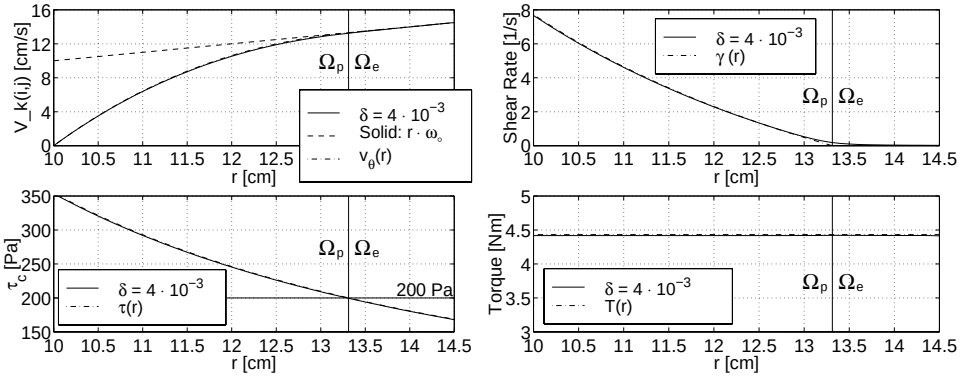


Figure 8.18: Computational analysis made at $\mathrm{i} \in [\mathrm{NX1} : \mathrm{NX2}] = [101 : 146]$ and $\mathrm{j} = 270$. The figure demonstrates the velocity profile V_k(i,j) (top left), shear rate $\dot{\gamma}_c$ (top right), shear stress τ_c (bottom left) and torque $\hat{T}_c$ (bottom right). The angular velocity for all illustrations is $\omega_o = 1\,\mathrm{rad/s}$. Dashed dotted lines represent the analytical results: $v_\theta(r)$, $\dot{\gamma}(r)$, $\tau(r)$ and $\hat{T}(r)$.

Figure 8.17 demonstrates isoplot of the computed von Mises shear stress τ_c, when the angular velocity is $\omega_o = 1\,\mathrm{rad/s}$. The isoline of $\tau_c = \tau_o = 200\,\mathrm{Pa}$ demonstrates the location of the boundary between the solid state Ω_e and the viscoplastic state Ω_p. This boundary is also shown with a dashed dotted line in Figure 8.16 for the velocity profile V_k(i,j).

Figure 8.18 shows computational analysis made at $(\mathrm{i}, \mathrm{j}) = ([\mathrm{NX1} : \mathrm{NX2}], 270) = ([101 : 146], 270)$. Calculating the solution error at these points, in the same manner as described in Equation 8.1, gives $RMS_{\mathbf{ac}} = 0.0028$. This value could have been anticipated from Table 8.2, given the similar conditions shown there. By using Equation 3.32 (Page 66), the correct location of the boundary between the plastic- and the solid state is calculated to be $R_s = 13.31\,\mathrm{cm}$. The value of $\tau_c(R_s) = 199\,\mathrm{Pa}$, not far away from the desired value of $\tau_o = 200\,\mathrm{Pa}$. This demonstrates that the computed

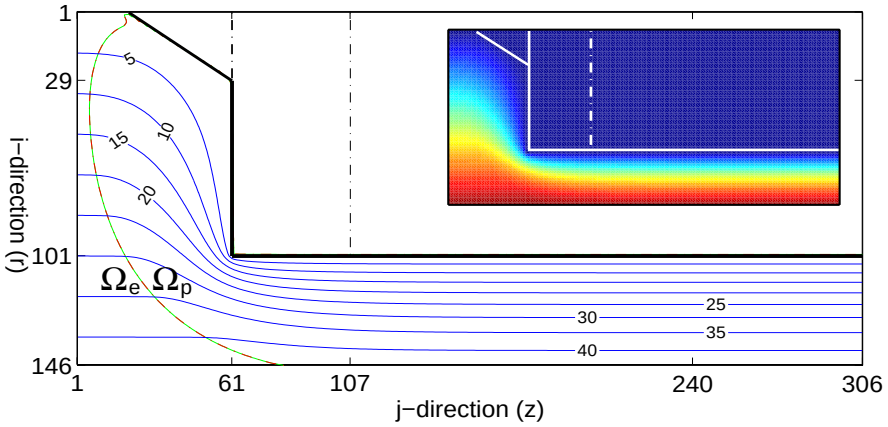


Figure 8.19: Velocity profile V_k(i,j) when $\omega_o = 3\,\mathrm{rad/s}$. The isolines start at $5\,\mathrm{cm/s}$ near the inner cylinder and with equal increments of $5\,\mathrm{cm/s}$, ends at $40\,\mathrm{cm/s}$ near the outer cylinder.

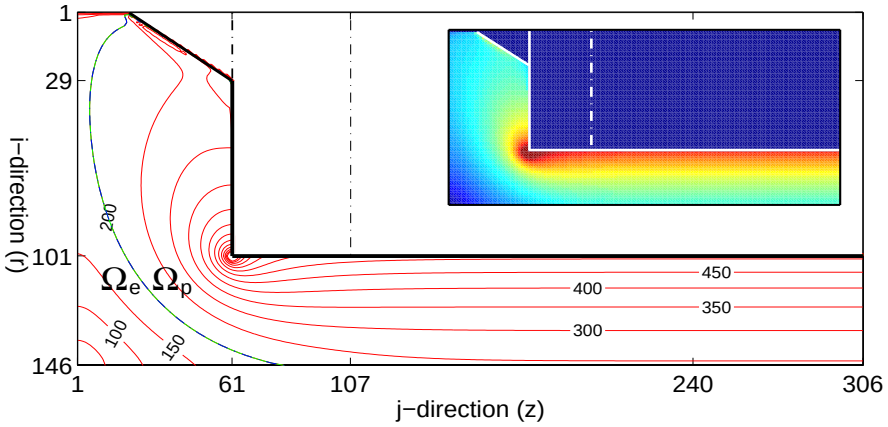


Figure 8.20: Shear stress τ_c when $\omega_o = 3\,\mathrm{rad/s}$. The isolines start at $50\,\mathrm{Pa}$ near the outer cylinder and with equal increments of $50\,\mathrm{Pa}$, ends at $800\,\mathrm{Pa}$ around the corner of inner cylinder.

von Mises shear stress of $\tau_c = \tau_o$ with $\delta = 4 \cdot 10^{-3}\,\mathrm{s}^{-1}$ is a good representation of the correct location of the boundary between the solid and the viscoplastic state.

The analytical torque $\hat{T} = 4.432\,\mathrm{Nm}$ is calculated from Equation 3.24, using $(R_i, R_s, h) = (10, 13.31, 19.9)\,\mathrm{cm}$. This equation assumes that the velocity v_θ is not changing with z (i.e. assumes z-independence). The numerical torque, calculated in the software, with full z-dependency, gives $\hat{T}_c = 4.425\,\mathrm{Nm}$. Comparing the two values, results in a $\Delta\hat{T} = 100 \cdot (\hat{T} - \hat{T}_c(R_i, h))/\hat{T} = 0.16\%$ difference. Comparing the analytical torque with the numerical torque $\hat{T}_c(R_i) = 4.417\,\mathrm{Nm}$, however now only using the velocity information at $\mathrm{j} = 270$ (i.e. assuming z-independence), gives $\Delta\hat{T} = 100 \cdot (\hat{T} - \hat{T}_c(R_i))/\hat{T} = 0.33\%$.

Figures 8.19 and 8.20 demonstrates isoplot of the computed velocity profile V_k(i,j) and the computed von Mises shear stress τ_c, when the angular velocity ω_o has been increased up to $3\,\mathrm{rad/s}$. With the increased angular velocity, it is apparent that the solid domain Ω_e has been reduced as a result of the concomitant increase in the computed von Mises shear stress τ_c.

8.7.2 ConTec Viscometer 4

The viscometric values used in this section, consist of $\tau_o = 60\,\text{Pa}$ and $\mu = 10\,\text{Pa}\cdot\text{s}$. The regularization parameter has the same value as previously, namely $\delta = 4{\cdot}10^{-3}\,\text{s}^{-1}$. Figure 8.21 demonstrates isoplot of the computed velocity V_k(i,j) when the rotational frequency f_o is 0.1 rps. The dashed dotted line demonstrates the (approximate) location of the boundary between the solid state Ω_e and the viscoplastic state Ω_p. As is evident, the solid region Ω_e is rather small relative to the whole solution domain Ω. The reason for this is due to the slimmer viscometric geometry of the ConTec Viscometer 4: As is evident with Equation 3.30 (Page 65), the plug starts at the outer cylinder (R_o), when the angular velocity ω_o is low as $\omega_o^p = \omega_o|_{R_s=R_o} = (\tau_o/\mu)\cdot 0.03$. However, for the ConTec **BML** Viscometer 3 this value is equal to $\omega_o^p = (\tau_o/\mu)\cdot 0.18$. Hence, a plug state is less likely to occur for the former viscometer, which results in the relatively smaller solid domain Ω_e.

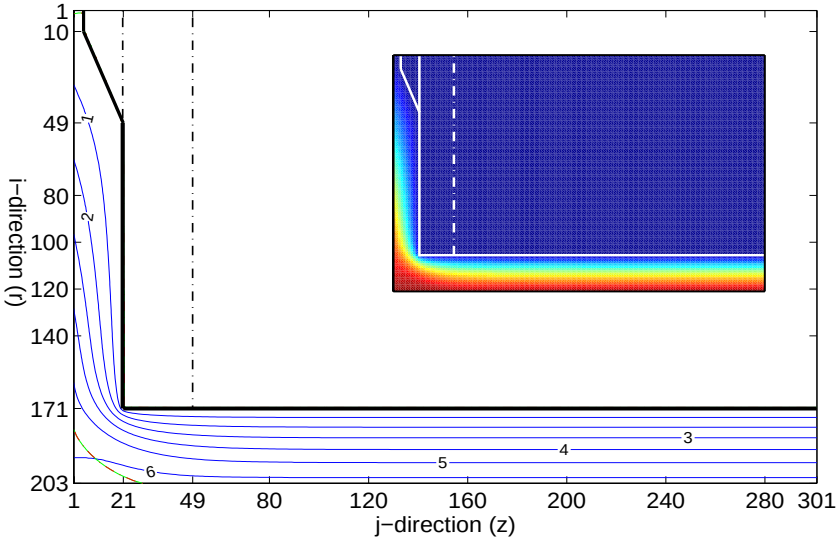


Figure 8.21: Velocity profile V_k(i,j) when $f_o = 0.1\,\text{rps}$. The isolines start at $1\,\text{cm/s}$ near the inner cylinder and with equal increments of $1\,\text{cm/s}$, ends at $6\,\text{cm/s}$ near the outer cylinder.

As mentioned in Section 8.7.1, when considering a suspension like mortar and cement paste, one has to realize that the smooth velocity profile shown in Figure 8.21 does not represent the velocity profile for the individual solid particles. This is because such type of motion is on a scale below what can be provided by Equation 2.18. The velocity profile of the solid particles is much more discontinuous and random than represented in this figure. Instead, Figure 8.21 shows the velocity profile of CPs with the CM coordinates at the corresponding spatial points $\mathbf{r}^{CM} \equiv \mathbf{x}(\mathbf{X}, t) = (r, \theta, z, t)$. As shown with Equation 2.6 (Page 13), this type of velocity consists of the mass averaged velocity of all the solid particles composing the CP. Therefore, a spatial point in the figure represents the average velocity of all the solid particles at and surrounding the point (r, z). As such, any random and discontinuous velocity contribution from the individual solid particles are summarized out, resulting in the smooth velocity profile shown in this figure.

Figure 8.22 demonstrates isoplot of the computed von Mises shear stress τ_c, when

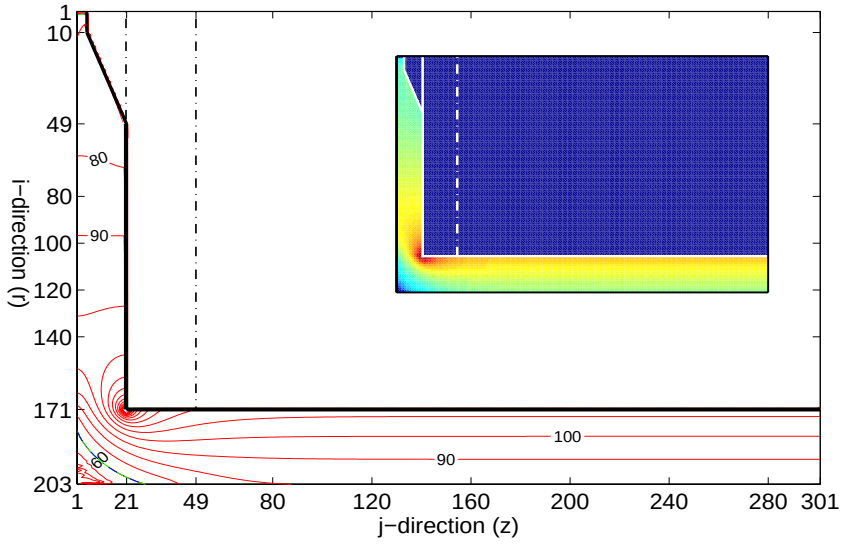


Figure 8.22: Shear stress τ_c when $f_o = 0.1\,\text{rps}$. The isolines start at $30\,\text{Pa}$ near the outer cylinder and with equal increments of $10\,\text{Pa}$, ends at $200\,\text{Pa}$ around the corner of inner cylinder.

the rotational frequency f_o is $0.1\,\text{rps}$. As before, the isoline of $\tau_c = \tau_o = 60\,\text{Pa}$ demonstrates the (approximate) location of the boundary between the solid state Ω_e and the viscoplastic state Ω_p. This boundary is also shown with a dashed dotted line in Figure 8.21 for the velocity profile V_k(i,j).

Figure 8.23 shows computational analysis made at $(i, j) = ([NX1 : NX2], 270) = ([171 : 203], 270)$. Calculating the solution error at these points, in the same manner as described in Equation 8.1, gives $RMS_{\mathbf{ac}} = 0.0019$. By using Equation 3.32 (Page 66), the correct location of the boundary between the plastic- and solid state is calculated to be $R_s = 11.38\,\text{cm} > R_o$ and hence no plug is occurring at the top region of the viscometer $\Rightarrow R_s = R_o = 10.1\,\text{cm}$.

The analytical torque $\hat{T} = 0.652\,\text{Nm}$ is calculated from Equation 3.24, using

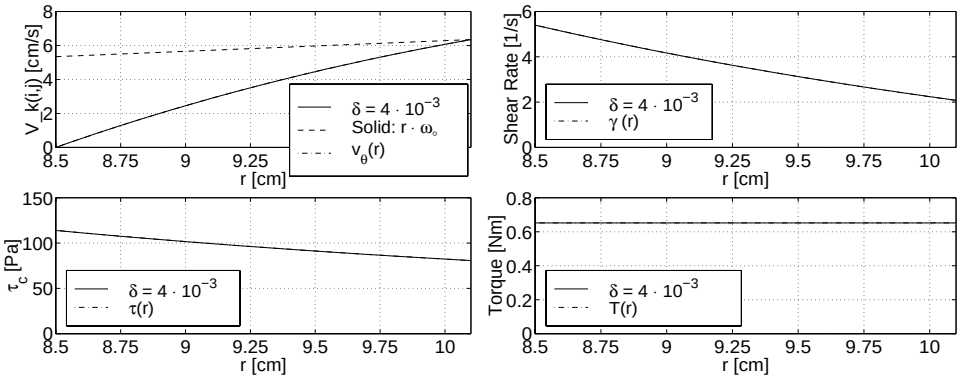


Figure 8.23: Computational analysis made at $i \in [NX1 : NX2] = [171 : 203]$ and $j = 270$. Same description applies here as for Figure 8.18.

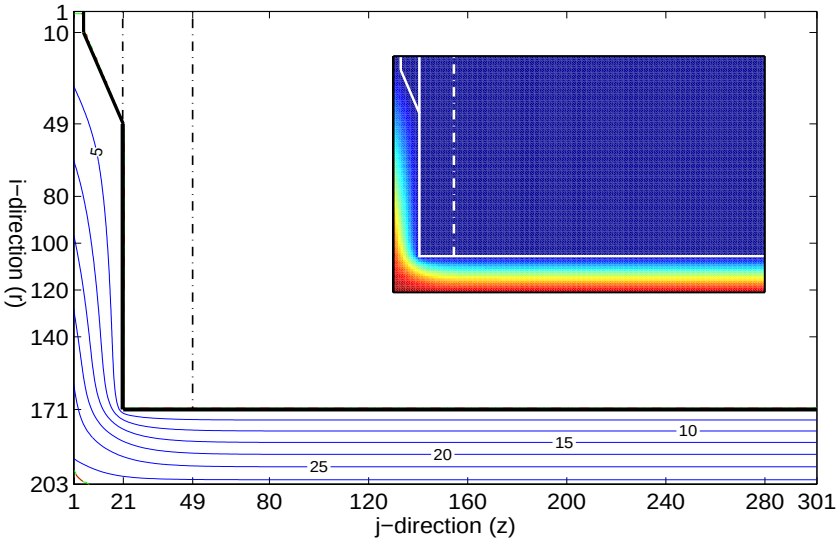


Figure 8.24: Velocity profile V_k(i,j) for the ConTec Viscometer 4 when $f_o = 0.5$ rps. The isolines start at 5 cm/s near the inner cylinder and with equal increments of 5 cm/s, ends at 30 cm/s near the outer cylinder.

$(R_i, R_o, h) = (8.5, 10.1, 12.6)$ cm. As mentioned earlier, this equation assumes that the velocity v_θ is not changing with z (i.e. assumes z-independence). The numerical torque, calculated in the software, with full z-dependency, gives $\hat{T}_c = 0.654$ Nm. Comparing the two values, results in a $\Delta\hat{T} = 100 \cdot (\hat{T} - \hat{T}_c(R_i, h))/\hat{T} = -0.31\%$ difference. The numerical torque generated by only using the velocity information at j = 270 (i.e. assuming z-independence), gives $\hat{T}_c(R_i) = 0.652$ Nm. This value is equal to the analytical torque $\hat{T}$.

Figures 8.24 and 8.25 demonstrates isoplot of the computed velocity V_k(i,j) and the computed von Mises shear stress τ_c, when the rotational frequency f_o has been increased up to 0.5 rps. With the increased angular velocity, it is apparent that the solid domain Ω_e has reduced as a result of the concomitant increase in the computed von Mises shear stress τ_c. Almost the entire solution domain Ω, now consist of the viscoplastic state Ω_p.

8.8 Summary

With Sections 8.4, 8.5 and 8.6 it was demonstrated that the numerical software Viscometric-ViscoPlastic-Flow produces accurate results, when simulating a viscoplastic fluid. In some parts, this demonstrations is based on comparison of numerical results with known analytical results of Section 3.3. For example, this was done when verifying numerical convergence. But also, the accuracy of the software was demonstrated with known mathematical tools: Issues like convergence, consistency and (apparent) stability has been dealt with.

It is clear with the parallel isolines in Figure 8.16 to Figure 8.25 and from the few torque calculations, that at the upper part of either ConTec **BML** Viscometer 3 or ConTec Viscometer 4, a z-independence can be assumed as is done when deriving

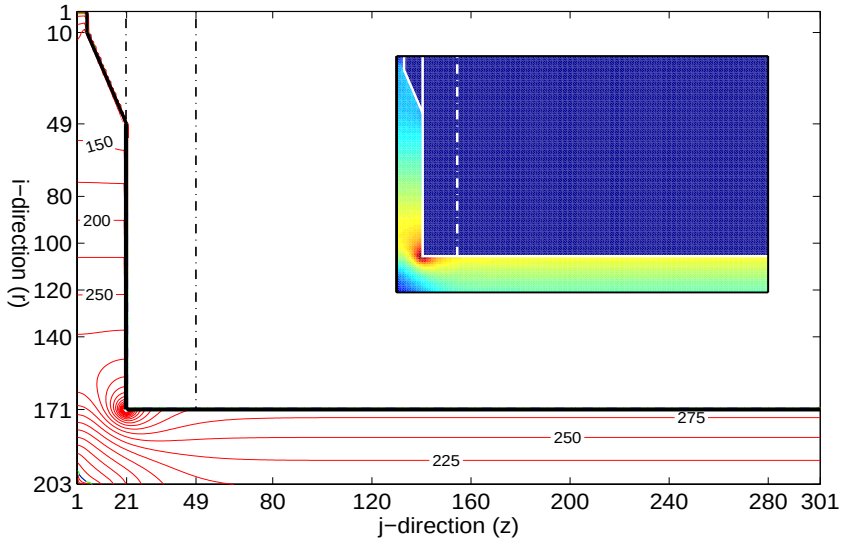


Figure 8.25: Shear stress τ_c for the ConTec Viscometer 4 when $f_o = 0.5\,\text{rps}$. The isolines start at $25\,\text{Pa}$ near the outer cylinder and with equal increments of $25\,\text{Pa}$, ends at $600\,\text{Pa}$ around the corner of inner cylinder.

the well-known **Reiner-Riwlin** equation (see Section 3.3.1 on Page 56).

In Chapter 9, the memory modules $\tilde{\Gamma}$ and $\tilde{\Theta}$ will be used (see Section 7.10). This means that the shear viscosity η will gain a memory of the recent events. However, such a maneuver does not come without a cost. In activating the memory modules, the calculation time increases to some large extent. This is not only due to the increased number of calculations necessary at each time step, but also because of reduced time step Δt used in maintaining stability. Fortunately, with the demonstration of z-independence, it is no longer necessary to include the bottom part of the solution region $\Omega \cup \partial\Omega$. In deleting it, the Neumann boundary condition $\partial v_\theta / \partial z = 0$ is applied at the new bottom part of the solution domain. The implementation of this condition is done in the same manner as is done for the upper part, shown in Section 7.7 (see Pages 164 and 186). With this scheme, the calculation time is much reduced, making a larger amount of numerical investigations possible.

Chapter 9

Thixotropic Explorations of Viscoplastic Fluid

9.1 Introduction

Background

The interest in thixotropic phenomena is nearly as old as modern rheology [75]. An increasing number of real materials has been found to show these effects. Also, they have been applied in various industrial applications. The term thixotropy was originally coined to describe an isothermal reversible, gel-sol (i.e. solid-liquid) transition due to mechanical agitation [75]. In a textbook by Barnes et al. [9], thixotropy is defined as "*A decrease of the apparent viscosity under constant shear stress or shear rate, followed by a gradual recovery when the stress or shear rate is removed. The effect is time-dependent.*" (with *apparent viscosity* it is meant the shear viscosity η).

The amount of theoretical literature on the above-mentioned time-dependent material is limited [9], however, there is a comprehensive review article about the subject done by Mewis [75]. In this paper, the various approaches used to measure thixotropy is represented. For example, one approach mentioned is to measure the torque $\hat{T}$ under a linear increase and then decrease in the rotational frequency f_{o}. If the test sample is thixotropic, the two torque curves produced do not coincide, causing rather a hysteresis loop. The area of a loop can then be used as a measure of the degree of thixotropy. While hysteresis loops are useful as a preliminary indicator of behavior, they do not provide a good basis for quantitative treatments [128]. However, attempts can be made to quantify the thixotropic behavior with such torque curves, by their integration [128, 6, 7]. Another approach possible in studying thixotropic behavior, is by monitoring the decay of measured torque from an initial value $\hat{T}_{\mathrm{o}}$ to an equilibrium value $\hat{T}_{\mathrm{e}}$ with time t, at a constant rotational frequency f_{o} [128]. In some cases, a simple exponential relationship can be found, but other and more complicated relationship can also exist. Lapasin et al. [67], makes the use of this approach, using a three different types of functions, however more complicated than the simple exponential form. Nevertheless, extracting shear viscosity parameters by this approach is very limited because the shear viscosity equation can only be valid when only one constant rotational frequency is applied for each experiment. Considering a shear viscosity equation extracted from such experiment and then using it in another

experiment of much more complicated shear history, will render almost impossible. As will be demonstrated in this chapter, for the latter case, such an equation must depend on fading memory information of both the shear rate $\dot{\gamma}$ (related to dispersion rate) and of the coagulation rate H.

In the article by Mewis [75], different theoretical approaches are mentioned to create a comprehensive thixotropic shear viscosity equation: **1**) By the use of axiomatic theory of continuum mechanics, **2**) by using the theory of structural kinetics, **3**) by starting from the microstructure to calculate the rheological behavior and **4**) by using thermodynamic arguments to develop thixotropic theories. In this chapter, a combination of first and second approach is used. The axiomatic theory consists of using Equation 9.7 (Section 9.3.2), while the theory of structural kinetics is extracted from the modified Hattori-Izumi theory of Section 9.3.1. The validity of the approach used here, is verified by reproducing the measured torque $\hat{T}$ by numerical means, under a very complex shear rate conditions.

Combination of Experimental and Numerical Approach

The underlying mathematics of the software VISCOMETRIC-VISCOPLASTIC-FLOW is presented in Chapter 7. In Chapter 8, the accuracy of the results produced by the software, is demonstrated to be of a good quality. In the current chapter, this software is used in a complex thixotropic analysis made on some cement pastes. Basically the analysis consists of reproducing the measured torque $\hat{T}$ by numerical means and in doing so, extracting a more detailed shear viscosity function η for the test material.

The objectives of this chapter is to test the Hattori-Izumi theory (Section 2.4.2) and its modification (Section 9.3). In doing this, one is also testing if the thixotropic behavior of cement paste is governed by a combination of reversible coagulation, dispersion and re-coagulation of the cement particles. In addition to this, the functional form of the shear viscosity function $\eta = \eta(\mathbf{x}, t)$ extracted in this chapter, is of great interest. Finally, the purpose of this chapter is to produce supplementary information about the effects of the polymers used in this thesis. The (super)plasticizers used in the cement pastes are the **VHMW Na** (Section 9.4), **HMW Na** (Section 9.6), **HMW Ca** (Section 9.7) and **SNF** (Section 9.8). Their characteristics are described in Section 4.2.2.

The viscometer used is the CONTEC VISCOMETER 4, hence its geometry is used when making the numerical simulations. Here, the shear viscosity η must gain a (fading) memory of the recent events. However, in making such mathematical maneuver, the calculation time increases to some large extent. This is not only due to the increased number of calculations necessary at each time step, but also because of reduced time step Δt used in maintaining stability. Fortunately, the bottom part of the solution domain $\Omega \cup \partial\Omega$ can be deleted without compromising accuracy. This approach is in accordance with the findings presented in Section 8.8. In doing this, the calculation time is decreased significantly. Also, the spatial step $\Delta r = \Delta z = 1\,\text{mm}$ is used instead of $\Delta r = \Delta z = 0.5\,\text{mm}$, to decrease the calculation time still further. Using a larger spatial step than 1 mm results in an inaccurate outcome. The effect of grid spacing on numerical results, when simulating a viscoplastic fluid, has been previously discussed in Section 8.6.1. Unless otherwise stated, the regularization parameter δ is always kept at $5 \cdot 10^{-3}$ in this chapter.

For one particular case, namely when using the **VHMW Na** polymer, the bottom part of the solution domain $\Omega \cup \partial\Omega$, is included in the numerical analysis. The results are presented in Section 9.5. There the findings of Section 8.8 is verified.

9.2 Experimental Setup

9.2.1 Mix Design

The water-cement ratio used for the cement paste is $w/c = 0.3$. The amount of plasticizer used in each batch is 0.5% sbwc. The volume of the test sample is about three liters, meaning that about 4.80 kg (1.54 l) of NORCEM STANDARD CEMENT and 1.44 kg of water is used in each batch.

The initial phase volume is equal to $\Phi = (1.54\,\mathrm{l})/(2.99\,\mathrm{l}) = 0.52$ and it is always increasing due to the growth of the cement particles and the reduction in free water, as a result of their chemical reactions (see also Footnote 33, Page 48).

The density is calculated to be $\rho = (4.80\,\mathrm{kg} + 1.44\,\mathrm{kg})/(2.99\,\mathrm{l}) = 2.09\,\mathrm{kg/l} = 2090\,\mathrm{kg/m^3}$. When assigning density in the source code **param.f90**, it is the SI value of 2090 that must be used.

9.2.2 Mixing Procedure

The fact that the test sample consist of only pure cement paste, results in a reduced reproducibility. Formation of some small coagulated cement clumps, during the mixing process of water and cement, has a serious effect on measured viscometric values.

At the start of mixing, pouring the whole 1.44 liter of water within 60 seconds, into the 4.80 kg of cement, was shown to be catastrophic, creating coarsely coagulated cement clumps, hard to brake apart. To avoid this, the water is rather added[1] in small portions for three minutes, while mixing at speed 1 (the speed settings are described in Section 5.3.1). Hand mixing is also important in braking up slam layers and the remaining coagulated cement clumps that resided at the bottom of the mixing bowl. Previously, with mortars and concrete, such clump formation was of lesser concern, since the gravel- and sand particles served as a grinding- and dispersing agents.

After some trial and error, the **mixing procedure** for the cement paste became as listed below.

Mixing procedure for cement paste.

1. $t \in [0\ \mathrm{min}, 3\ \mathrm{min}]$: Mixing of cement and water at speed one. [As described in Section 5.2.1, the water is premixed with lignosulfonate (or with **SNF**)]. Most of the cement particles are (more or less) moistened within the first 60 seconds.
2. $t \in [3\ \mathrm{min}, 6\ \mathrm{min}]$: Hand mixing and resting.
3. $t \in [6\ \mathrm{min}, 10\ \mathrm{min}]$: Mixing at speed 2 (mixing at speed 3, resulted in too much splashing in most cases).
4. $t \in [10\ \mathrm{min}, 11\ \mathrm{min}]$: Check with hand mixing, if the cement suspension is homogeneous.

Still, the above mixing procedure only solved what is visible to the naked eye. As will be apparent, for example in Figure 9.8, the issue of bad reproducibility is never completely resolved. To solve (or rather reduce) this problem, three repeated batches are mixed and tested, for each type of plasticizer used. The average of the three results is then used to represent rheological behavior of the test material. For example, in Figure 9.3 (or in Figure 9.8), the dotted lines represent each of the three batches, the solid line demonstrates the average of these measurements and the double line demonstrates the numerical simulation result.

[1] The cement is first placed into the mixing bowl and then the water is added.

9.2.3 Measuring Procedure

At 12, 42, 72 and 102 minutes after the initial water addition, rheological measurements with the ConTec Viscometer 4 is performed. Immediately after each measurement, a remixing by hand is done to ensure homogeneous mixture. Right afterwards the measuring unit[2] is submerged into the cement paste and stays in this position until the next measurement is finished. Hence, a complete rest applies for the test material prior to the coming measurement. The resting consisted of about 29 minutes and was considered to be sufficient time for the test sample to gain a completely coagulated state, at the start of the next measurement: $U_3|_{t=0} \equiv U_o = 1$. No remixing, with the Hobart mixer, applies between measurements. Since the objectives is to investigate thixotropic effects of the cement suspension, it would be pointless to use the Hobart mixer to disperse the coagulated cement particles and brake up the structure that the viscometer is supposed to measure.

9.2.4 Rotational Frequency f_o

The boundary condition that apply in the numerical simulation must always be the same as in the real experiment. This means that the rotational frequency $f_o|_{exp}$ that the viscometer generates for the experiment, must be equal to the rotational frequency $f_o|_{num}$ used in the numerical simulation $\Rightarrow f_o = f_o|_{exp} = f_o|_{num}$. This condition applies here, and the rotational frequency used in either case is shown in the left illustration of Figure 9.1. It is the source code **motion.f90** that provides the rotational frequency in the calculation and it is constructed with a series of exponential functions. Using a series of *arctan* functions was also tested, but resulted in a worse representation of the actual rotational frequency.

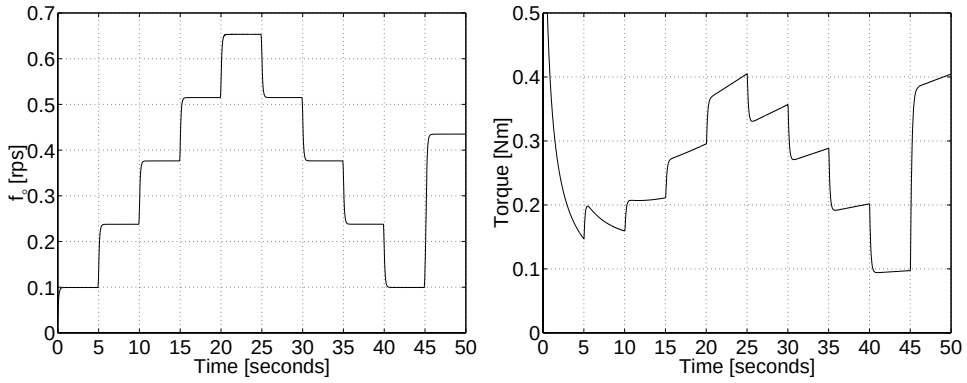


Figure 9.1: To the left: Rotational frequency that is used in both the numerical simulation and in the experiment. To the right: Torque curve $\hat{T}_c$ calculated in the software ViscometricViscoPlastic-Flow, using Equations 9.1 and 9.2.

Referring to the left illustration of Figure 9.1: In the time period between 0 and 25 seconds the *stepwise increasing shear rate sequence* is applied. This is followed with the *stepwise decreasing shear rate sequence* in the time period between 25 and 45 seconds.

[2]The measuring unit is the inner cylinder as shown with the center illustration of Figure 3.1.

9.3 Modified Hattori-Izumi Theory

The viscosity of Hattori and Izumi is given by Equation 2.27 and is reproduced below.

$$\eta_{\mathrm{HI}} = B_3 \, J_{\mathrm{t}}^{2/3} \tag{9.1}$$

As mentioned in Section 2.4.2, the term $J_{\mathrm{t}} = n_3 \, U_3$ represents the number of (reversible) junctions between primary cement particle that can undergo a reversible coagulation (see the paragraph just below Equation 2.64, Page 49). [Both U_3 and J_{t} are also referred as the reversible coagulation state (see Section 2.6.3)]. The number of (reversible) junctions is calculated by Equation 2.45 and is reproduced with Equation 9.2, below.

$$J_{\mathrm{t}} \equiv J_3 = n_3 \, U_3 = \frac{n_3 \, [U_{\mathrm{o}}(\dot{\gamma} \, H \, t^2 + 1) + H \, t]}{(H \, t + 1)(\dot{\gamma} \, t + 1)} \tag{9.2}$$

The variable H is the coagulation rate coefficient, described in Section 2.5.3. As is mentioned in Sections 2.4.2, the term $J_{\mathrm{t}}|_{t=0} \equiv J_{\mathrm{o}} = n_3 \, U_{\mathrm{o}}$ describes the number of (reversible) junctions between the primary cement particles (that can undergo a reversible coagulation), at the beginning of an experiment $t = 0$. For each batch tested, a number of four experiments are made, namely at 12, 42, 72 and 102 minutes after initial water addition. Each experiment is made with the ConTec Viscometer 4 and begins at the (experimental) time $t = 0$ and ends at $t = 50\,\mathrm{s}$. This time variable is not to be confused with the time duration from mixing of water and cement clinker. The former time period spans only over 50 seconds as shown in Figure 9.1 to the left, while the latter spans over the whole 102 minutes as discussed in Section 9.2.3.

The immediate problem with Equation 9.2 becomes apparent when taking the limit $t \to \infty$. In such calculation, the condition $J_{\mathrm{t}} = n_3 \, U_{\mathrm{o}}$ is produced, regardless of the values H or $\dot{\gamma}$ that apply; i.e. under sufficiently large value of t, the time factor will overshadow all physical processes described with the coagulation rate constant H and the shear rate $\dot{\gamma}$. A consequence of such unphysical characteristics, is shown in the right illustration of Figure 9.1. This figure demonstrates an example of the computed torque $\hat{T}_{\mathrm{c}}$, using Equations 9.1 and 9.2 without any modifications. Suffice to say, using a broad range of numerical values for B_3, n_3, U_{o} and H (also those suggested by the original authors), it became impossible to reproduce the measured torque $\hat{T}$.

In the following three sections, a description is made of some necessary modifications applied to Equations 9.1 and 9.2. Much of these modifications are based on (semi) empirical considerations, rather than of theoretical ones. They are made with the sole purpose of being able to calculate a torque $\hat{T}_{\mathrm{c}}$ that is identical to the measured torque $\hat{T}$ (applied on the inner cylinder, from the test material). The computed torque is calculated with $\hat{T}_{\mathrm{c}} = 2 \, \pi \, R_{\mathrm{i}}^2 \, h \, \eta \, \dot{\gamma}_{\mathrm{c}}$. This equation is extracted by using Equations 2.23 (Page 17) and 7.4 (Page 156) when integrating Equation 3.16 (Page 58). The shear rate $\dot{\gamma}$ is calculated according to Equation 7.3 (Page 156), and shear viscosity η according to Equation 9.7.

9.3.1 First Modification: Fading Memory

Equation 9.2 is not far from being correct. By replacing the two terms $\dot{\gamma} \, t$ and $H \, t$ with the memory modules $\tilde{\Gamma}$ and $\tilde{\Theta}$, a better result starts to emerge. The last-mentioned

two modules are defined by Equations 9.3 and 9.4 (these are the same equations as presented in Section 7.10).

$$\tilde{\Gamma}(r,z,t) = \int_0^t \alpha(t-t')\ \dot{\gamma}(r,z,t')\ dt' \tag{9.3}$$

$$\tilde{\Theta}(r,z,t) = \int_0^t \beta(t-t')\ H(\dot{\gamma},r,z,t')\ dt' \tag{9.4}$$

The two terms α and β are memory functions to be defined shortly. For the reasons mentioned in Section 7.10, the integration must begin from the start of an experiment $t' = 0$ to the current time $t' = t$ (where $t \in [0, 50\,\mathrm{s}]$). With the above modification, the current (reversible) coagulation state $U_3 = J_\mathrm{t}/n_3$ is now controlled, not only by the current coagulation $H\,t$ and dispersion $\dot{\gamma}\,t$, but also by their past history. This is shown with the equation below.

$$U_3 = \frac{J_\mathrm{t}}{n_3} = \frac{U_\mathrm{o}(\tilde{\Gamma}\,\tilde{\Theta}+1)+\tilde{\Theta}}{(\tilde{\Theta}+1)(\tilde{\Gamma}+1)} \tag{9.5}$$

The issue of in what manner the shear viscosity η should memorize the past, is set by the memory functions α and β. Here, they are defined with exponential function, meaning that the shear viscosity will remember more the recent events, relative to the far past. Such type of memeory is called **fading memory**. The memory functions are given by Equation 9.6.

$$\alpha(t-t') = e^{-(t-t')/m_\mathrm{a}} \quad \wedge \quad \beta(t-t') = e^{-(t-t')/m_\mathrm{b}} \tag{9.6}$$

The terms m_a and m_b determine how much the shear viscosity η should memorize the past and as such, they have the physical unit of seconds. Often, the term m_a is set equal to 30 seconds, meaning that the shear viscosity η will have a good recollection about the shear rate $\dot{\gamma}$ for the past 30 seconds with increasing amnesia beyond that time point. Using $m_\mathrm{a} \to \infty \wedge m_\mathrm{b} \to \infty$ will result in $\alpha(t-t') = \beta(t-t') = 1$, meaning that the shear viscosity will always have a full recollection about its past.

9.3.2 Second Modification: Yield Value

As shown with Equation 9.1, the possibility for a yield value is discarded. However, in the quest of reproducing the measured torque $\hat{T}$ by numerical means, it became necessary to include such a term into the shear viscosity function. This resulted in the temporary shear viscosity function of $\eta_\mathrm{tmp} = a_1\,\eta_\mathrm{HI} + a_2\,\eta_\mathrm{HI}/\dot{\gamma}$. Both $a_1\,\eta_\mathrm{HI}$ and $a_2\,\eta_\mathrm{HI}$ are related to the increased [decreased] momentum exchange between the cement particles with increasing [decreasing] number of reversible junctions J_t. The relationship between the reversible coagulation state J_t and the shear viscosity η, is summarized in Section 2.6.3 (see also illustration **D** in Figure 2.11, Page 27). In a fully dispersed state, $J_\mathrm{t} = n_3\,U_3 = 0$, both terms $a_1\,\eta_\mathrm{HI}$ and $a_2\,\eta_\mathrm{HI}$ are zero. This means that the temporary shear viscosity η_tmp becomes also zero. However, as discussed in Section 2.4.1, in a fully dispersed state ($J_\mathrm{t} = 0$) the cement particles will still interact with each other by collisions (i.e. by momentum exchange), that will result in a certain minimum and non-zero shear viscosity. This is more specifically shown with illustrations **A** and **B** in Figure 2.11 (see also Figures 2.9 and 2.10). With this

in mind, it is only natural to include two constant terms into the shear viscosity equation, which takes such contribution into account. That is, the shear viscosity now consist of $\eta = \eta_{\text{tmp}} + (\mu + \tau_{\text{o}}/\dot{\gamma}) = (\mu + a_1\,\eta_{\text{HI}}) + (\tau_{\text{o}} + a_2\,\eta_{\text{HI}})/\dot{\gamma}$. More precisely, with the these modifications, the shear viscosity becomes as shown with Equation 9.7.

$$\eta = B + \frac{\sqrt{C_{\text{y}}}}{\dot{\gamma}} \tag{9.7}$$

The viscometric values B and $\sqrt{C_{\text{y}}}$ will be referred as the **effective** (or actual) **plastic viscosity** and the **effective** (or actual) **yield value**, respectively. They are given by Equations 9.8 and 9.9, shown below.

$$B = \mu + (a_1\,B_3\,n_3^{2/3})\,U_3^{2/3} = \mu + \tilde{\mu} \tag{9.8}$$

$$\sqrt{C_{\text{y}}} = \tau_{\text{o}} + (a_2\,B_3\,n_3^{2/3})\,U_3^{2/3} = \tau_{\text{o}} + \tilde{\tau}_{\text{o}} \tag{9.9}$$

The role that the two viscometric terms μ and τ_{o} play in Equation 9.7, is somewhat more complicated than mentioned above. This is because of how it is possible to categorize two types of coagulation. First is the **reversible coagulation**, meaning that the coagulated cement particles can disperse from each other, for the given available power $\dot{w}$ to the suspension (see Equation B.34, Page 390); i.e. the power $\dot{w}$ is equal to or greater than the work required to continuously separate cement particles, that are glued together by the energy barrier $V_{\text{T}}^{\max} + |V_{\text{T}}^{\min}|$ (see Figure 2.19, Page 42). This power will become more and more insufficient in dispersing all coagulated cement particles because of how the potential energy barrier $V_{\text{T}}^{\max} + |V_{\text{T}}^{\min}|$, that keeps the cement particles glued together in a coagulated state, is increasing in depth (see the DLVO theory on Page 41). The depth of $V_{\text{T}}^{\max} + |V_{\text{T}}^{\min}|$ will increase as a result of the chemical reactions between the cement particles and water, and possibly also because of some damage to the adsorbed polymer, at the same time. When this barrier has gained such a depth that available power $\dot{w}$ cannot brake two coagulated cement particles apart, then these particles have entered a **permanent coagulation**. This is the second type of coagulation.

At first consideration, the two terms μ and τ_{o} describes only the minimum value that the shear viscosity η can have when the cement particles are fully dispersed; i.e. these two terms describes a certain basic minimum momentum transfer between the cement particles, which the process of coagulation and dispersion is not part of. This is shown with illustrations **A** and **B** in Figure 2.11 (Page 27). However, with permanent coagulation (see illustration **C** in Figure 2.11), these basic interactions becomes larger, resulting in a certain contribution to μ and τ_{o}. The number of permanent junctions J_{t}^{p} that is created in the process, is not a part of the "bookkeeping" made by Equation 9.5. This is because of how the theory of Section 2.4.2 does not account for the possibility of permanent coagulation[3]. As such, the calculated value of U_3 (by Equation 9.5) is only relative to reversible coagulated cement particles. From one experiment (say at 12 min) to the next (at 42 min), more and more cement particles becomes permanently coagulated, i.e. the value of J_{t}^{p} increases steadily during the time period of 29 minutes. This means that the total number of (primary) cement particles, that can undergo a reversible coagulation (designated with n_3), is decreasing from one experiment to the next. For each experiment that spans over only 50 seconds, it can be assumed that very few cement particles undergo a permanent coagulation (i.e. the value of

[3]This is apparent since the number of reversible junctions J_{t} can theoretically always go back to zero, while experimentally the total number of junctions $J_{\text{t}}^{\text{tot}} = J_{\text{t}}^{\text{p}} + J_{\text{t}}$ (in the suspension) can not.

J_t^P is roughly a constant). This can be assumed because of how both μ and τ_o are a constant over the short time interval of 50 seconds, while a measurement is being conducted. This means that the number of reversible primary particles n_3 is (more or less) a constant during a single experiment, as is assumed in Section 2.4.2 and hence in Equation 9.5. However, the value of n_3 is a different (or rather a reducing) constant, from one experiment to the next.

It is rather during the time period of 29 minutes, between each experiment, that the number of permanent junctions J_t^P are gradually increasing, giving a gradual increase in either or both μ and τ_o. As shown with Equation 2.65, the number of reversible junctions J_t are also gradually increasing during this same time period, resulting in an increase for both $\tilde{\mu}$ and $\tilde{\tau}_o$ (or equally, contributing to the initial value $J_o = n_3 U_o$ that will be used in the coming experiment). Coagulation rate H during this period of 29 minutes, is given by Equation 2.59 (Page 46). During a single measurement (that spans over 50 seconds), both $\tilde{\mu}$ and $\tilde{\tau}_o$ are rapidly changing because of the dispersion and the (reversible) re-coagulation of cement particles. Coagulation rate H during this short period of time, is then given by Equation 2.61, or more precisely by Equation 9.10 shown shortly.

9.3.3 Third Modification: Coagulation Rate

In the Hattori-Izumi theory, it is assumed that the coagulation rate H is constant, only dependent on Brownian motion (i.e. on temperature T). As mentioned in Section 2.5.3, such an assumption cannot hold for an orthokinetic[4] conditions. The stirring ($\dot{\gamma} \neq 0$) causes the cement particles to be thrown together at a larger rate than the normal diffusion rate, and hence the orthokinetic processes increases coagulation. In Section 2.5.3, it is pointed out that orthokinetic coagulation process plays an important role in determining the coagulation rate H for the cement particles and hence in determining the correct coagulated state U_3. From this, it is clear that the coagulation rate coefficient is dependent on stirring $H = H(\dot{\gamma})$.

Since the stirring causes the cement particles to be thrown together at a larger rate than the normal diffusion rate, it is to be expected that the condition $dH(\dot{\gamma})/d\dot{\gamma} \geq 0$ applies $\forall\ \dot{\gamma} \geq 0$. However, it can be argued that with continuously increasing shear rate $\dot{\gamma}$, the mutual kinetic energy of two cement particles that are about to collide, will also increase in the process. There will come a condition (designated with $\dot{\gamma}_{cr}$) where this kinetic energy is sufficient to overcome the potential energy barrier $V_T^{max} + |V_T^{min}|$ that can hold them together in a coagulated state. The rate of collisions between the cement particles will always increase with increasing shear rate $\dot{\gamma}$. However, beyond the point of $\dot{\gamma}_{cr}$, fewer and fewer of these collisions will result in actual coagulation. This means that the coagulation rate will start to decrease with further increase in shear rate $\Rightarrow dH/d\dot{\gamma} \leq 0\ \forall\ \dot{\gamma} \geq \dot{\gamma}_{cr}$. Hence the overall characteristics of the coagulation rate function could be anticipated as $dH/d\dot{\gamma} > 0\ \forall\ \dot{\gamma} < \dot{\gamma}_{cr}$ and $dH/d\dot{\gamma} \leq 0\ \forall\ \dot{\gamma} \geq \dot{\gamma}_{cr}$. An example of a function that hold such a properties would be of an exponential form: $H = H_o\, e^{-(\dot{\gamma}-\dot{\gamma}_{cr})^2}$.

In this thesis, several functions where tested, like of $H = a\,\dot{\gamma}^b$ or $H = a\,e^{-(\dot{\gamma}-b)^2}$ ($a > 0 \wedge b > 0$). However, after a large number of trials in reproducing the measured torque in the best manner, the coagulation rate function had to have the functional form described with Equation 9.10. This function is rather peculiar, because it holds the condition $dH/d\dot{\gamma} \leq 0\ \forall\ \dot{\gamma} \geq 0$. With the above text in mind, this is only the half

[4] **Orthokinetic** $\Rightarrow$ with stirring $\Rightarrow \dot{\gamma} \neq 0$ (**Perikinetic** $\Rightarrow$ no stirring $\Rightarrow \dot{\gamma} = 0$).

of the characteristics that was to be expected to be necessary in the calculations.

$$H(\dot{\gamma},t) = \frac{K(t)}{\dot{\gamma}^2 + l} \quad \forall \quad t > 0 \quad \wedge \quad H(\dot{\gamma},0) = \frac{k_1\,(1 - U_{\mathrm{o}})}{4\,l} \tag{9.10}$$

The term l in the above equation, is an empirical constant, kept equal to $l = 1\,\mathrm{s}^{-2}$ at all times. The function $K(t)$ is an empirical step-function that increases and decreases parallel to the angular velocity $\omega_{\mathrm{o}}(t)$ of the outer cylinder. In the time period from 0 to 25 seconds after the start of measurements, the angular velocity $\omega_{\mathrm{o}}(t)$ is increasing in steps as shown with the left illustration of Figure 9.1. In this period the function K holds the value k_1. From 25 to 45 seconds, the angular velocity $\omega_{\mathrm{o}}(t)$ is decreasing, and hence the value of K is changed to k_2. Between the time 45 to 50 seconds, the angular velocity is increased again, resulting in $K = k_3$. Summarizing the above text with an equation, gives the following:

$$K(t) = \begin{cases} k_1 = 0.005\,\mathrm{s}^{-3} & \text{if} \quad d\omega_{\mathrm{o}}(t)/dt \geq 0 \quad \Rightarrow \quad 0 \leq t < 25\,\mathrm{s} \\ k_2 = 0.1\,\mathrm{s}^{-3} & \text{if} \quad d\omega_{\mathrm{o}}(t)/dt \leq 0 \quad \Rightarrow \quad 25 \leq t < 45\,\mathrm{s} \\ k_3 = 0.005\,\mathrm{s}^{-3} & \text{if} \quad d\omega_{\mathrm{o}}(t)/dt \geq 0 \quad \Rightarrow \quad 45 \leq t \leq 50\,\mathrm{s} \end{cases} \tag{9.11}$$

Unless otherwise stated, the values of k_1, k_2 and k_3 remain as shown above.

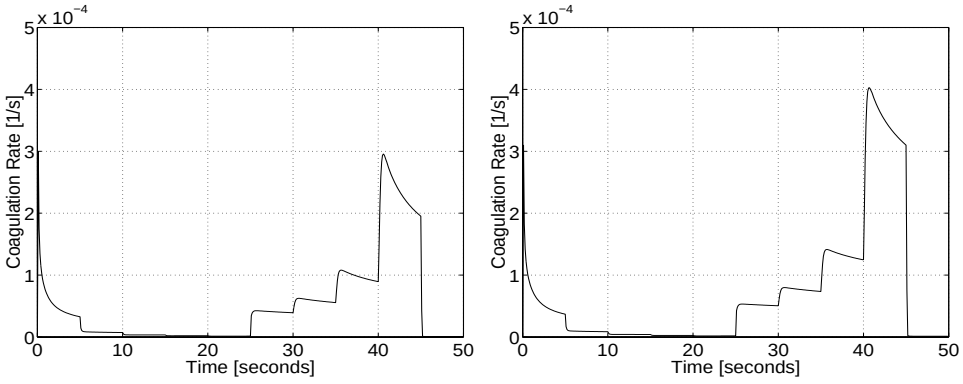


Figure 9.2: Calculated coagulation rate function $H = H(\dot{\gamma},t)$ that exists at the inner cylinder $r = R_{\mathrm{i}}$, using Equations 9.10 and 9.11. To the left is the result produced when using **HMW Na** ($t = 72\,\mathrm{min}$), while to the right is the result produced when using **HMW Ca** ($t = 42\,\mathrm{min}$).

Figure 9.2 demonstrates a plot of the coagulation rate function $H = H(\dot{\gamma},t)$ that is calculated at the inner cylinder ($r = R_{\mathrm{i}}$), using Equations 9.10 and 9.11. Hattori and Izumi report values of coagulation rate constant ranging from $H = 10^{-9}\,\mathrm{s}^{-1}$ to $H = 10^{-2}\,\mathrm{s}^{-1}$, depending on the mix design of the suspension in question [42, 41]. As shown in Figure 9.2, the value of the coagulation function is below $5 \cdot 10^{-4}\,\mathrm{s}^{-1}$ at the inner cylinder ($r = R_{\mathrm{i}}$). Interestingly, this range is within the values reported by Hattori and Izumi. However, to reproduce the measured torque in the best manner, the largest value that the coagulation function must hold in this thesis, is $H_{\max} = k_2/l = 10^{-1}\,\mathrm{s}^{-1}$. This maximum value frequently appears near the outer cylinder ($r = R_{\mathrm{o}}$), where the shear rate is at its lowest value $\dot{\gamma} \approx 0$. Hence, the coagulation rate function $H(\dot{\gamma},t)$ can hold a value that is an order of magnitude larger than the maximum value reported.

9.4 VHMW Na

Figure 9.3 demonstrates both the experimental results given by the ConTec Viscometer 4 and the numerical simulation results given by the software Viscometric-ViscoPlastic-Flow, using the values in Table 9.1. Four illustrations are presented, marked with "$t = 12\,\mathrm{min}$", "$t = 42\,\mathrm{min}$", "$t = 72\,\mathrm{min}$" and "$t = 102\,\mathrm{min}$" and represents the time from the initial water addition. The dotted line represents all experimental results of the three repeated batches. The solid line demonstrates the average of these measurements and the double line demonstrates the numerical result.

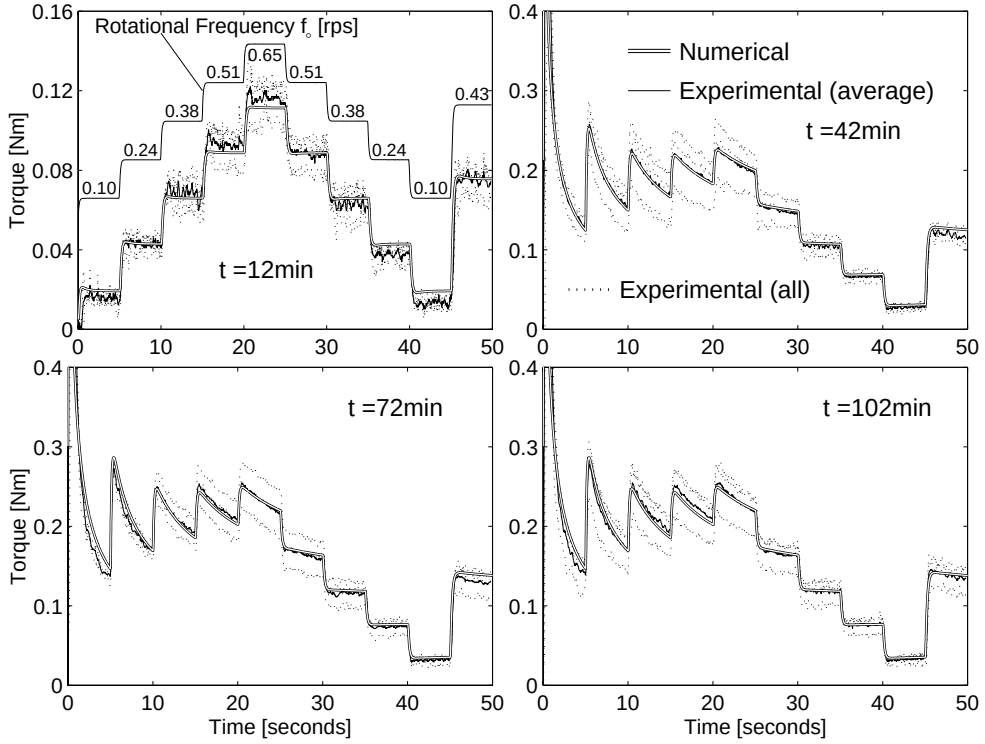


Figure 9.3: Measured- and computed torque $\hat{T} \wedge \hat{T}_{\mathrm{c}}$, for a cement paste using **VHMW Na**.

Table 9.1: Viscometric parameters used in the numerical simulations.

VHMW Na	μ	$a_1\, B_3\, n_3^{2/3}$	τ_{o}	$a_2\, B_3\, n_3^{2/3}$	U_{o}	m_{a}	m_{b}
	[Pa · s]	[Pa · s]	[Pa]	[Pa]	-	[s]	[s]
$t = 12\,\mathrm{min}$	0.70	15	0	0	$5 \cdot 10^{-3}$	1	0
$t = 42\,\mathrm{min}$	0.60	30	0	12	1	30	0
$t = 72\,\mathrm{min}$	0.65	33	0	24	1	30	0
$t = 102\,\mathrm{min}$	0.65	33	0	24	1	30	0

As shown in Figure 9.3, there is an excellent correspondence between the measured torque $\hat{T}$ and its numerical counterpart $\hat{T}_{\mathrm{c}}$. This applies for all four illustrations. As mentioned above, the viscometric values used in calculating the torque $\hat{T}_{\mathrm{c}}$ is shown in Table 9.1. All the values in this table are extracted by the means of "trial and

error". That is, all the values are systematically changed until the computed torque can completely overlap the measured torque (this is a rather time consuming process). There are two issues about this method of trial and error that should be clear to the reader:

1. Very few variables are allowed to change from one experiment to the next. They are at the most seven, as shown in Table 9.1. Many of these free variables do not have to be changed at all, from one experiment to the next. This means that the degree of freedom is even lower than indicated by the pre-mentioned table. It is rather the physics of Section 9.3 that controls the computed torque behavior (however, through the few values of Table 9.1).

2. The same computed torque curve (or rather solution $\mathbf{v} = v_\theta(r, z, t)\,\mathbf{i}_\theta$) cannot be extracted by two different sets of viscometric values $[\mu,\ \tau_\mathrm{o},\ U_\mathrm{o},\ m_\mathrm{a},\ m_\mathrm{b},\ldots]$ $= [a, b, c, d, e, \ldots]$ and $[\mu,\ \tau_\mathrm{o},\ U_\mathrm{o},\ m_\mathrm{a},\ m_\mathrm{b},\ldots] = [x, y, z, u, v, \ldots]$. This became quite clear in the process of the above-mentioned trial and error, where the different viscometric values in Table 9.1, gave a very specific and different computed torque response. This behavior is also a requirement for well-posed problems; i.e. the theory of existence, uniqueness and stability for a solution of PDE, require this kind of behavior (see for example the textbook by Strauss [117]). Hence, the values shown in Table 9.1 are not just a coincidence, but rather correspond to the cement paste rheological response. This is perhaps more apparent in Section 9.9.4, when summarizing all such data in this chapter into one figure. There, some familiar trends from Chapter 6 reappears.

The above two points applies also for Table 9.2 (**HMW Na**), Table 9.3 (**HMW Ca**) and Table 9.4 (**SNF**), shown in the coming sections.

In reproducing the measured torque in the best fashion, the coagulation-memory m_b is set equal to zero for all experiments. According to Equations 9.6 and 9.4, this means that β and therefore $\tilde{\Theta}$ are also equal to zero. Physically, with $m_\mathrm{b} = 0$ it is assumed that no re-coagulation is occurring during a single experiment of 50 seconds. That is, the shear viscosity is not allowed to remember that a re-coagulation is occurring, since apparently no such re-coagulation is taking place to begin with.

A very similar condition applies for the dispersion-memory m_a, for the experiment at 12 minutes after water addition. With the very effective dispersing mechanism of the **VHMW Na** polymer and with the very recent agitation from the initial mixing inside the Hobart mixer (see Section 9.2.2), the cement particles are very much dispersed at the start of the experiment. As such, the value of m_a must be small, because the shear viscosity cannot[5] be allowed to remember a dispersion that cannot take place to begin with. With the pre-mentioned condition of $\tilde{\Theta} = 0$, no re-coagulation is either occurring and hence no dispersion of newly formed coagulated cement particles, can either occur.

In the source code **viscous.f90**, the amnesia in coagulation is enforced not by using $m_\mathrm{b} = 0$, but rather by implementing $\tilde{\Theta} = 0$ directly.

The left illustration of Figure 9.4 demonstrates[6] how the (reversible) coagulation state U_3 is uniformly dropping as a function of time at all spatial points and is a direct consequence of the condition $\tilde{\Theta} = 0$ that applies. The particular result shown

[5]The dispersion cannot take place because most of the cement particles are already dispersed.

[6]It should be noted that the first U_3 value shown in the figure, is always just below 1. This is because it applies at 0.01 second after start of the experiment. At the start of the experiment ($t = 0\,\mathrm{s}$), the value of U_3 always hold the value of U_o, which is equal to 1 in this case.

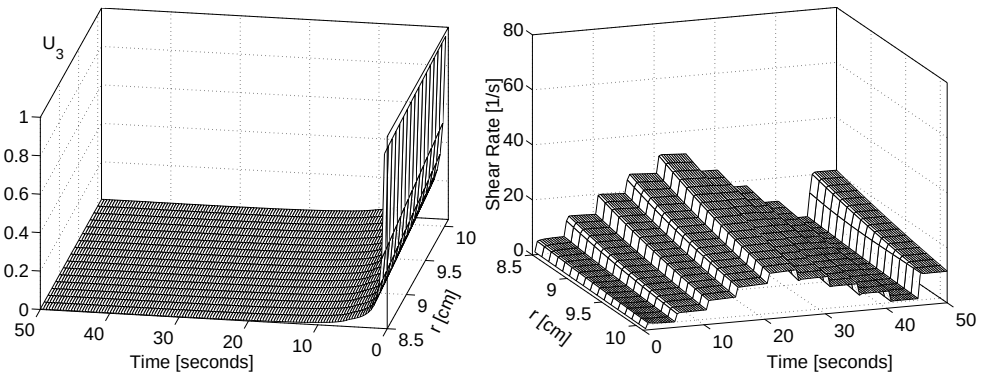


Figure 9.4: The coagulation state U_3 (to the left) and computed shear rate $\dot{\gamma}_c$ (to the right) as a function of radius r and time t, when using **VHMW Na** ($t = 72\,\text{min} \wedge t = 102\,\text{min}$).

in this figure, apply for the experiment conducted at 72 and 102 minutes after water addition. Very similar result is produced after 42 minutes.

For the experiment conducted at 12 minutes after water addition, the initial coagulation state U_o cannot be set equal to 1. This is so because of the very recent agitation from the initial mixing inside the Hobart mixer (see Section 9.2.2). The initial value of $U_o = 5 \cdot 10^{-3}$ in Table 9.1, is not determined by theoretical means, but rather it is determined (in the same manner as all the other viscometric values) in the process of reproducing the measured torque in the best manner.

In Table 9.1, it is interesting to note how little evolution there is in the viscometric values as a function of time from water addition. The greatest difference exists between the measurement conducted at 12 minutes after water addition and the rest of the measurements, conducted at 42, 72 and 102 minutes after water addition. The yield value τ_o is always equal to zero and the plastic viscosity μ is approximately a constant, equal to $0.65\,\text{Pa} \cdot \text{s}$. The same behavior applies when using the same type of polymer, in mortar case of [0.6% sbwc; $T_o = 23°\text{C}$; $w/c = 0.4$; OPC], shown in Section 6.4.6.

Summary of Main Results

1. The good results shown in Figure 9.3, demonstrate that the theory of Section 9.3 comes a long way in describing the mechanism occurring inside the cement paste.

2. No re-coagulation was needed (during the experiment) in the numerical simulations shown in Figure 9.3 ($\tilde{\Theta} = 0$).

9.5 Height Dependency (VHMW Na)

As mentioned in the introductory part of this chapter, the bottom part of the solution domain $\Omega \cup \partial\Omega$ is deleted. According with the findings presented in Section 8.8, this can be done without compromising accuracy. Since the result in the last-mentioned section is based on steady state calculations only, it is natural to make the same type of analysis when a severe time dependency is present.

Figure 9.5 demonstrates the velocity field, when the whole solution domain $\Omega \cup \partial\Omega$ is used, in a time dependent calculation. This particular velocity field applies at the moment of $t = 11\,\mathrm{s}$ after start of the second experiment. The magnitude of the largest velocity vector shown (i.e. the largest cone) for this particular case, is $23.9\,\mathrm{cm/s}$.

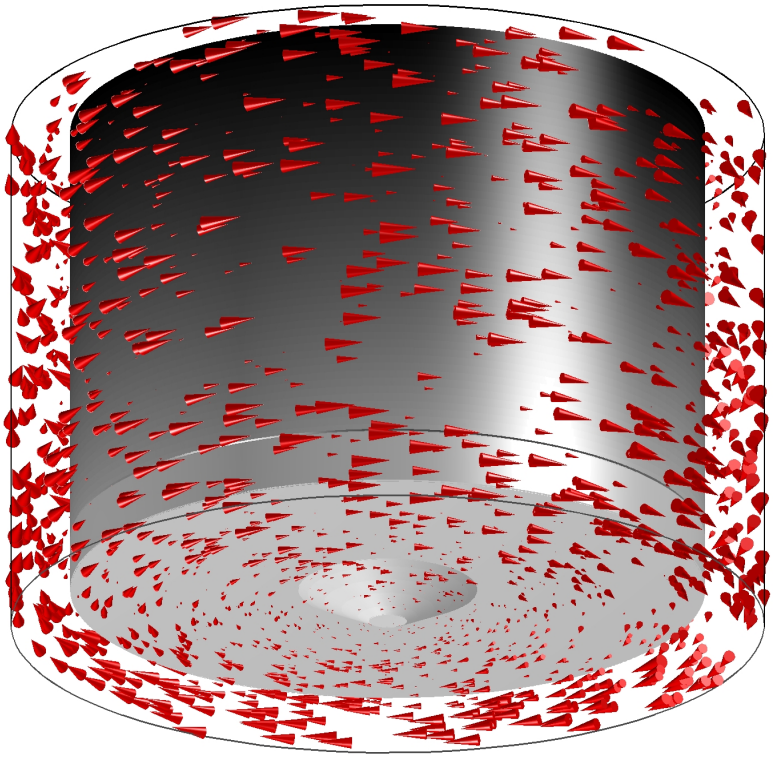


Figure 9.5: Three dimensional vector plot of velocity $\mathbf{v}$ at the moment of $t = 11\,\mathrm{s}$ after start of the second experiment. The whole solution domain $\Omega \cup \partial\Omega$ is used (**VHMW Na** $\wedge\ t = 42\,\mathrm{min}$).

As mentioned in Section 8.3.2, when the whole solution domain is included, the applied torque is calculated from the **j-grid** point NY2mH3 = 49 to NY2 = 281, with i = 171 (see Figure 8.2, Page 180). This torque will be designated here as $\hat{T}_{\mathrm{cA}}$ to distinguish from the torque $\hat{T}_{\mathrm{cB}}$ calculated when the bottom part of the solution domain $\Omega \cup \partial\Omega$ is deleted. The value of $\hat{T}_{\mathrm{cB}}$ is extracted from Figure 9.3.

Figure 9.6 demonstrates a comparison of the computed torque $\hat{T}_{\mathrm{cA}}$ and $\hat{T}_{\mathrm{cB}}$. The left illustration demonstrates the difference in torque $\hat{T}_{\mathrm{cA}} - \hat{T}_{\mathrm{cB}}$. As shown, the torque $\hat{T}_{\mathrm{cA}}$ is always larger than $\hat{T}_{\mathrm{cB}}$. This is likely because of the additional shear stresses that originates from the bottom plate of the viscometer. The right illustration of Figure 9.6 demonstrates the plot of all values, both $\hat{T}_{\mathrm{cA}}$ and $\hat{T}_{\mathrm{cB}}$. With this

comparison, it is evident that very little (or practically no) difference exists between torque values, generated with or without the bottom part of the solution domain $\Omega \cup \partial\Omega$ (i.e. with or without the bottom part of the viscometric geometry).

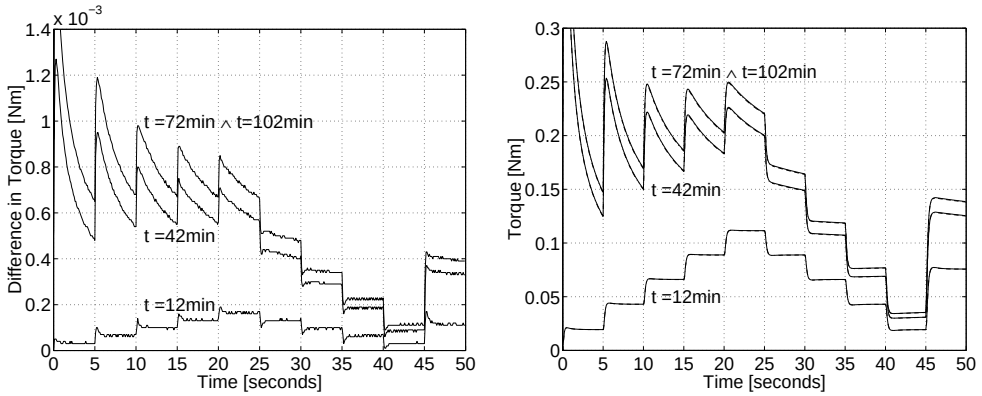


Figure 9.6: Comparison of calculated torque with ($\hat{T}_{\mathrm{cA}}$) and without ($\hat{T}_{\mathrm{cB}}$) the bottom part. The $\hat{T}_{\mathrm{cB}}$ values are extracted directly from Figure 9.3. To the left: Graphical presentation of the difference $\hat{T}_{\mathrm{cA}} - \hat{T}_{\mathrm{cB}}$. To the right: All values presented, both $\hat{T}_{\mathrm{cA}}$ and $\hat{T}_{\mathrm{cB}}$.

Figure 9.7 demonstrates the relative difference between $\hat{T}_{\mathrm{cA}}$ and $\hat{T}_{\mathrm{cB}}$. The values shown are calculated as $100 \cdot (\hat{T}_{\mathrm{cA}} - \hat{T}_{\mathrm{cB}})/\hat{T}_{\mathrm{cA}}$. As shown, the maximum difference is only around 0.4% to 0.5%.

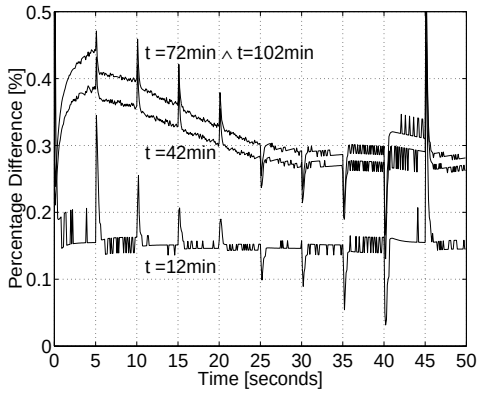


Figure 9.7: Relative difference between $\hat{T}_{\mathrm{cA}}$ and $\hat{T}_{\mathrm{cB}}$ (calculated as $100 \cdot (\hat{T}_{\mathrm{cA}} - \hat{T}_{\mathrm{cB}})/\hat{T}_{\mathrm{cA}}$).

Summary of Main Results

1. The bottom part of the solution domain $\Omega \cup \partial\Omega$ can be deleted without compromising accuracy.

2. This result is in an accordance with the findings presented in Section 8.8.

9.6 HMW Na

Figure 9.8 demonstrates both the experimental results given by the viscometer and the numerical results given by the software, using the values in Table 9.8. The same markings apply in this figure, as in Figure 9.3.

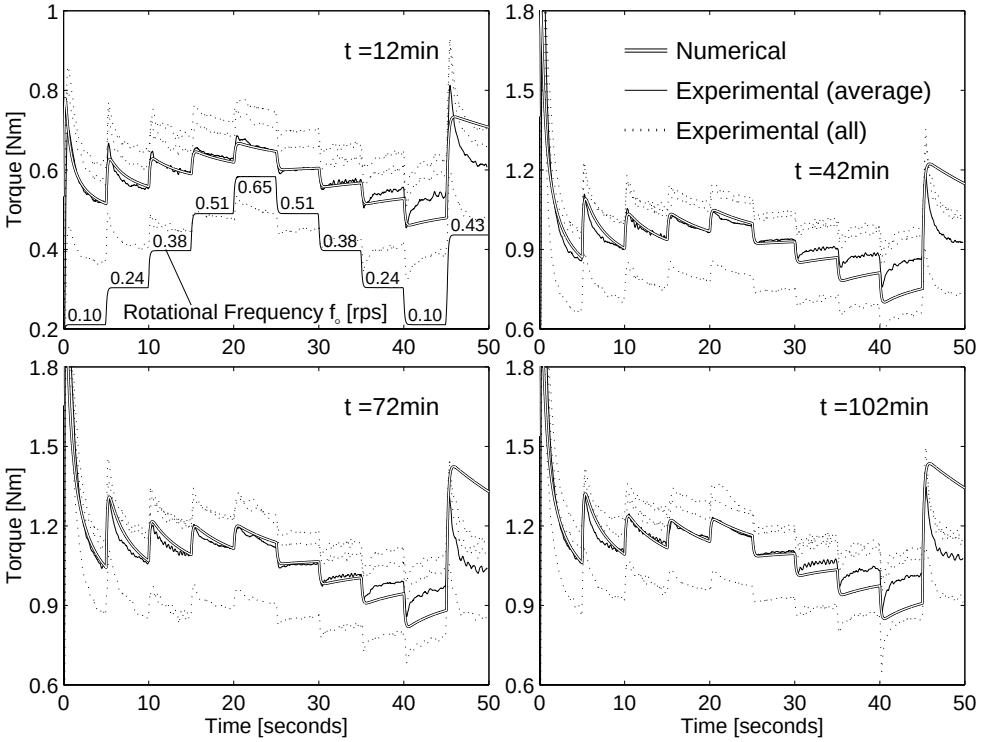


Figure 9.8: Measured- and computed torque $\hat{T} \wedge \hat{T}_{\mathrm{c}}$, for a cement paste using **HMW Na**.

Table 9.2: Viscometric parameters used in the numerical simulations.

HMW Na	μ	$a_1\, B_3\, n_3^{2/3}$	τ_{o}	$a_2\, B_3\, n_3^{2/3}$	U_{o}	m_{a}	m_{b}
	[Pa · s]	[Pa · s]	[Pa]	[Pa]	-	[s]	[s]
$t = 12\,\mathrm{min}$	0.8	40	70	0	0.4	25	15
$t = 42\,\mathrm{min}$	1.0	45	104	90	1	40	10
$t = 72\,\mathrm{min}$	0.8	60	124	100	1	40	10
$t = 102\,\mathrm{min}$	0.8	60	130	75	1	40	10

As shown in Figure 9.8, there is an excellent correspondence between the measured torque $\hat{T}$ and its numerical counterpart $\hat{T}_{\mathrm{c}}$ in the first 30 to 35 seconds. Thereafter, the two torque terms $\hat{T}$ and $\hat{T}_{\mathrm{c}}$ start to deviate from each other. In particular, in the time period between 35 and 45 seconds, the test material creates a larger "rebuild" than is anticipated with the theory of Section 9.3. Changing the viscometric parameter in such manner that the computed torque $\hat{T}_{\mathrm{c}}$ coincide with the measured torque $\hat{T}$ for this particular time period, was rendered impossible with the current theory. Several ideas have been tested to resolve the matter, but without any real success. These

ideas will be discussed in Section 9.7.

As is shown in Table 9.2, the "coagulation memory" m_{b} is no longer zero. This means that the memory module $\tilde{\Theta}$ is also no longer kept equal to zero. The necessity of using a non-zero m_{b} term, means that re-coagulation is occurring while the experiment is taking place.

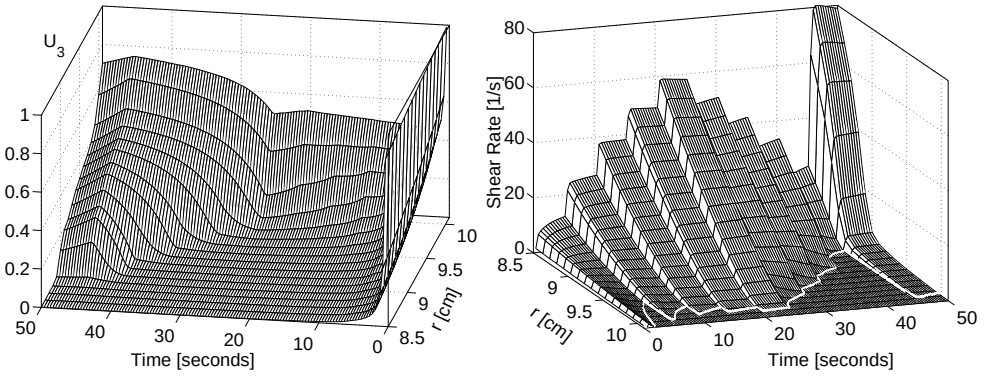


Figure 9.9: The coagulation state U_3 (to the left) and computed shear rate $\dot{\gamma}_{\mathrm{c}}$ (to the right) as a function of radius r and time t, when using **HMW Na** ($t = 72\,\mathrm{min}$).

The left illustration of Figure 9.9 demonstrates how the (reversible) coagulation state U_3 is changing rapidly as a function of time t and radius r. This is a different case than applies for Figure 9.4 previously. The increase in coagulation U_3 at different locations is a direct consequence of the condition $\tilde{\Theta} \neq 0$ that applies in the calculation. When considering how the coagulation (or kinetics of the structural rebuilding) is slower in comparison with that of dispersion (or structural breakdown) [67], the severe increase in U_3 between 30 and 45 seconds, could be unphysical and something that needs to be corrected. Several approaches will be considered in the later part of Section 9.7. The particular result shown in Figure 9.9, applies at 72 minutes after water addition. Only a slightly different profile of $U_3 = U_3(r, t)$ is produced for the experiment conducted after 42 and 102 minutes. This is (among other things) because of the small difference in shear rate $\dot{\gamma}$ produced as a result of the small difference in viscometric values used. A larger difference exists between the measurement at 12 minutes and the rest of the experiments, as is shown when comparing Figure 9.9 with Figure 9.10. The reason for the low coagulation state U_3 in the latter figure, is because of the low initial value U_{o} used.

As is shown in Table 9.2 (and also in Figure 9.10), for the experiment conducted at 12 minutes after water addition, the initial coagulation state was set equal to $U_{\mathrm{o}} = 0.4$. This is a somewhat larger value than used in Section 9.4, where the initial value was set equal to $U_{\mathrm{o}} = 5 \cdot 10^{-3}$. Here, the difference between the effectiveness of the **VHMW Na** over the **HMW Na** polymer, becomes evident. In both cases, the same recent agitation applies from the Hobart mixer (see Section 9.2.2). But still, as is evident with the value of U_{o}, a much larger coagulation state is factual for the **HMW Na**-case, at the start of the first experiment.

In Table 9.2, it is interesting to note how little evolution there is in the plastic viscosity $\mu \approx 0.85\,\mathrm{Pa \cdot s}$ as a function of time from water addition. The changes is more rapid for the yield value τ_{o}, where it is increasing from 70 to 130 Pa. This type of behavior is also observed when using the same type of polymer in mortar at

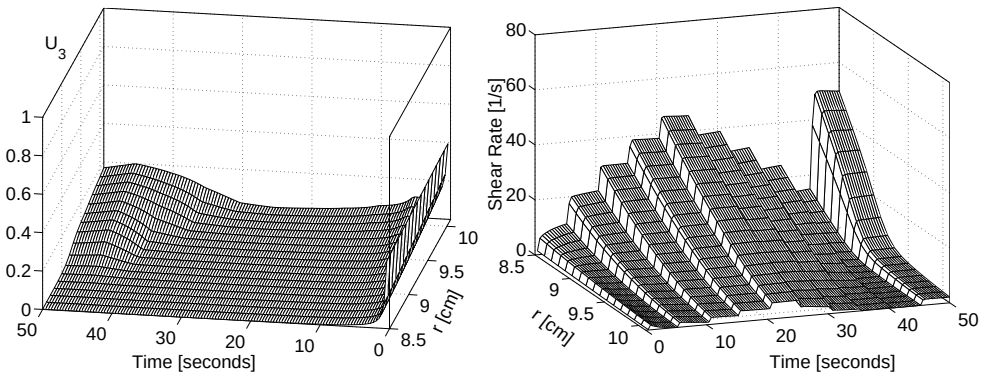


Figure 9.10: The coagulation state U_3 (to the left) and computed shear rate $\dot{\gamma}_c$ (to the right) as a function of radius r and time t, when using **HMW Na** ($t = 12\,\mathrm{min}$).

$w/c = 0.4$ (0.6 sbwc), shown in Section 6.2.1.

The right illustration of Figures 9.4, 9.9 and 9.10 demonstrates the computed shear rate $\dot{\gamma}_c$ as a function of radius r and time t. The white line on the $\dot{\gamma} = 0$ plane in Figure 9.9, demonstrates the boundary between the solid state Ω_e and the viscoplastic state Ω_p. It is localized by plotting the function $-\mathrm{II}_{\mathbf{S}}^{\mathrm{P}} = C_y$. More specifically[7], this boundary is localized by plotting $(\eta\,\dot{\gamma})^2 = C_y$, using Equation 9.7. When comparing these figures, one can see how great impact the shear viscosity η has on shear rate the $\dot{\gamma}$. With the different viscometric values applied (shown in Tables 9.1 and 9.2), the different shear rates $\dot{\gamma}$ becomes evident. Same type of conclusion is evident, when using a pure Bingham fluid in a steady state conditions, as is shown with Equation 3.28 (Page 59).

Summary of Main Results

1. A peculiar deviation exists between the computed torque $\hat{T}_c$ and the measured torque $\hat{T}$ between 35 and 50 seconds.

2. In present section, a re-coagulation (during the experiment) is needed to simulate the measured torque in the best fashion ($\tilde{\Theta} \neq 0$).

[7] $\Rightarrow -\mathrm{II}_{\mathbf{S}}^{\mathrm{P}} = (\mathbf{T} : \mathbf{T})/2 = (2\,\eta)^2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}/2 = \eta^2\,(2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}) = (\eta\,\dot{\gamma})^2$

9.7 HMW Ca

Figure 9.11 demonstrates both the experimental results given by the ConTec Viscometer 4 and the numerical simulation results given by the software Viscometric-ViscoPlastic-Flow, using the values in Table 9.3. The same markings apply in this figure, as in Figure 9.3.

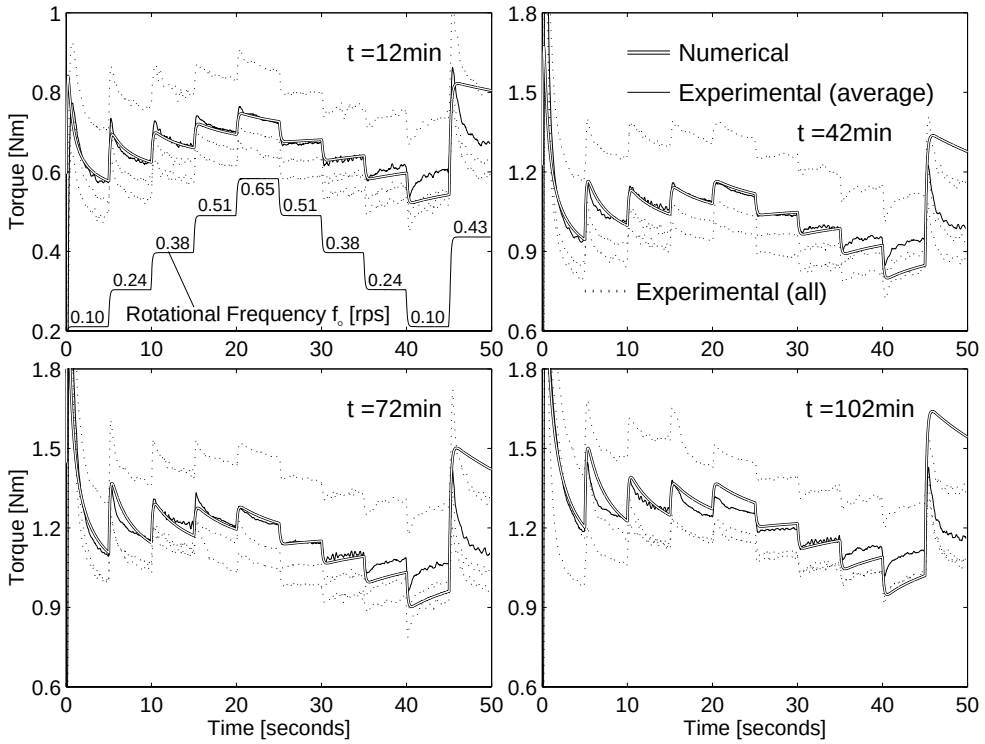


Figure 9.11: Measured- and computed torque $\hat{T} \wedge \hat{T}_{\mathrm{c}}$, for a cement paste using **HMW Ca**.

Table 9.3: Viscometric parameters used in the numerical simulations.

HMW Ca	μ	$a_1 B_3 n_3^{2/3}$	τ_{o}	$a_2 B_3 n_3^{2/3}$	U_{o}	m_{a}	m_{b}
	[Pa · s]	[Pa · s]	[Pa]	[Pa]	-	[s]	[s]
$t = 12\,\mathrm{min}$	0.8	40	80	0	0.4	15	30
$t = 42\,\mathrm{min}$	1.0	45	120	45	1	15	15
$t = 72\,\mathrm{min}$	0.5	60	140	50	1	15	15
$t = 102\,\mathrm{min}$	0.5	66	145	110	1	15	15

As applies for Figure 9.8, there is an excellent correspondence between the measured torque $\hat{T}$ and its numerical counterpart $\hat{T}_{\mathrm{c}}$ for the first 35 seconds. Thereafter, the two torque terms $\hat{T}$ and $\hat{T}_{\mathrm{c}}$ start to deviate from each other. As before, in the time period between 35 and 45 seconds, the test material creates a larger "rebuild" than is anticipated with the theory of Section 9.3, while between 45 and 50 seconds, the test material creates a larger "breakdown". This issue will be discussed shortly.

The time evolution of the effective (or the actual) plastic viscosity B and yield

value $\sqrt{C_y}$ (see Equations 9.8 and 9.9) for the four cases in Table 9.3, are shown in Figures 9.12 and 9.13. More precisely, these two figures demonstrates the time evolution (from water addition) of these two viscometric values as a function of (experimental) time t and radius r. The lowest "sheath" always applies at 12 minutes after water addition, and the remaining sheaths applies in chronological order, at 42, 72 and finally at 102 minutes, which consists of the highest sheath. From the two figures, it is interesting to note how the two values B and $\sqrt{C_y}$ are rapidly changing, both with location r and time t. The exception applies for the effective yield value at 12 minutes after water addition. There, this value is a constant, equal to $\sqrt{C_y} = \tau_o = 80\,\text{Pa}$ and therefore appears as a horizontal plane in Figure 9.13.

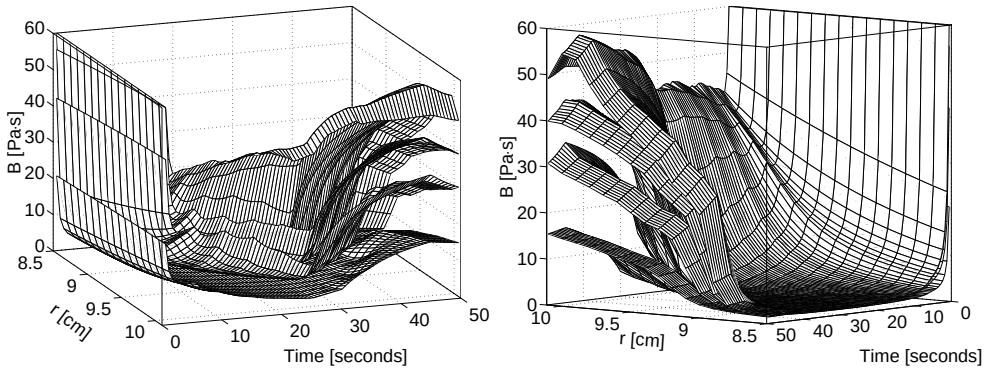


Figure 9.12: The effective plastic viscosity B as a function of radius r and time t. The two illustrations are of the same plot, however from two different angles. The polymer used consist of **HMW Ca** and the results shown, apply at 12, 42, 72 and 102 minutes after water addition.

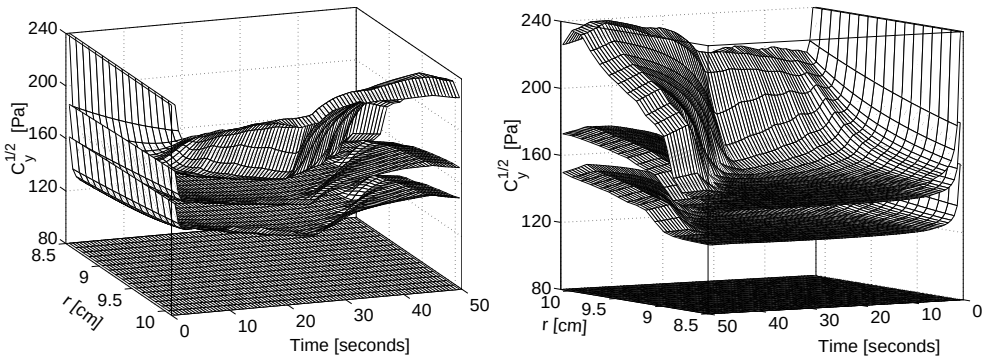


Figure 9.13: The effective yield value $\sqrt{C_y}$ as a function of radius r and time t. The two illustrations are of the same plot, however from two different angles. The polymer used consist of **HMW Ca** and the results shown, apply at 12, 42, 72 and 102 minutes after water addition.

Secondary Effect

A common characteristics in Figure 9.8 (**HMW Na**) and Figure 9.11 (**HMW Ca**) is apparent: In both cases, there is a deviation between the measured torque $\hat{T}$ and the computed torque $\hat{T}_c$. This deviation exists for most part in the time period between

35 and 50 seconds, and consist of two different types of deviations. The first type occurs in the time period between 35 and 45 seconds, where the test material creates a larger "rebuild" than is anticipated with the theory of Section 9.3. The second type occurs between 45 and 50 seconds. There the opposite problem exists, namely that the computed torque $\hat{T}_c$ fails to reproduce the same amount of "breakdown" as is measured. The latter problem can be solved, as is shown in the last part of this section. Unfortunately, the first problem remains unsolved in this thesis.

When calculating the number of connections J_t in Equation 9.5, it is assumed that the coagulated cement particles form a kind of open-chain-structure (i.e. a "string-of-pearls-structure"), rather than closely interconnecting spherical/cubical structure (with multiple internal cross-junctions). This assumption is implemented with Equation 2.35 (or Equation 2.28) on Page 31. Since the first 35 seconds of the experiment can be reproduced numerically, this assumption seems to be valid. That all experiments for the **VHMW Na**-case can be reproduced numerically, validates this assumption further. This is actually quite remarkable when considering how large the phase volume Φ actually is. Using $w/c = 0.3$, it is calculated to be 0.52 (see Section 9.2.1) and is always increasing from the time of water addition due to the chemical reactions ($\Phi \geq 0.52$). [It should be nevertheless clear that the modifications shown in Section 9.3, could implicitly account for a closely interconnecting spherical/cubical structure].

Equation 9.5 can also (roughly) be valid[8], when the coagulated cement particles form a "near-spherical-structure" (without multiple internal cross-junctions) as shown with illustration **I** of Figure 9.15. This means that regardless if the flocks of cement particles consist of the string-of-pearls-structure (illustration **A**), or of near-spherical-structure (illustration **I**), the same theory that counts the number of reversible junctions J_t between the cement particles, can be applied. However, for the given junction number $J_t = n_3 U_3$ and rotational frequency f_o, those two different structures could produce two different types of shear viscosity response η. With the near-spherical-structure, an additional viscosity contribution could be created, because of how free water is more immobilized (see illustration **I** in Figure 9.15). This results in a larger apparent phase volume Φ and thus gives a higher shear viscosity η than expected. This effect is somewhat discussed in a textbook by Barnes et al. [9]. Also, with increasing apparent phase volume Φ in an already packed suspension, some additional dilatancy effect (or "crowding" effect) could be a reality, that affects the shear viscosity η. The above idea is just a hypothesis. But regardless of the physical reason for the deviation between $\hat{T}_c$ and $\hat{T}$ in $35\,\mathrm{s} \leq t \leq 45\,\mathrm{s}$, the effect that is responsible for this, is designated here as the **secondary effect**.

The fact that the above-mentioned secondary effect is never present when using the **VHMW Na** polymer (Section 9.4), indicates that this effect is strongly related to the re-coagulation process that occurs in the end of each experiment. This is because of how experimentally, no rebuild in torque appears for the **VHMW Na**-case, and because of how numerically, the re-coagulation must be set equal to zero to imitate this measured torque.

As is shown with the arrow in the left illustration of Figure 9.14, the measured torque $\hat{T}$ in the time period from 40 to 45 seconds ($f_o = 0.1\,\mathrm{rps}$) gains almost the same equilibrium value as in the time period from 35 to 40 seconds ($f_o = 0.24\,\mathrm{rps}$). Actually this tendency is measured for all cases in Figures 9.8 and 9.11. This tendency is quite peculiar because in the former case, the rotational frequency f_o is less than

[8] Valid in the sense that Equation 2.35 (or Equation 2.28), more or less still apply.

half of the latter case (see Figure 9.1). It is not to be expected that coagulation per-se, is alone responsible for this large increase in measured torque. This is because of how coagulation (or kinetics of the structural rebuilding) is slower in comparison with that of dispersion (or structural breakdown) [67]; i.e. coagulation alone cannot produce the same rapid changes in torque $\hat{T}$ as dispersion does. Rather, the idea is that this large torque increase is a byproduct of coagulation, namely the above-mentioned **secondary effect**. More precisely, it can be suggested that some kind of new structure is forming as a result of the re-coagulation in the end of an experiment, when the shear rate $\dot{\gamma}$ starts to decrease. The result of this effect is sketched in the right illustration of Figure 9.14 and is marked as "Secondary Effects". For now, no explanation is given for the deviation that exists in the time period between 45 and 50 seconds, and is therefore marked with "Unknown Effect".

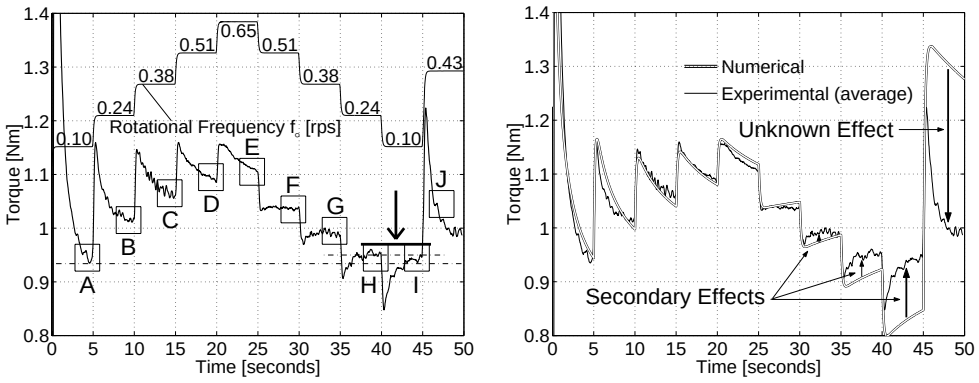


Figure 9.14: To the left: Measured torque $\hat{T}$, where the alphabetic markings corresponds to the ones shown in Figure 9.15. To the right: Illustration of the consequence for the secondary effect. (**HMW Ca** $\wedge\ t = 42\,\mathrm{min}$).

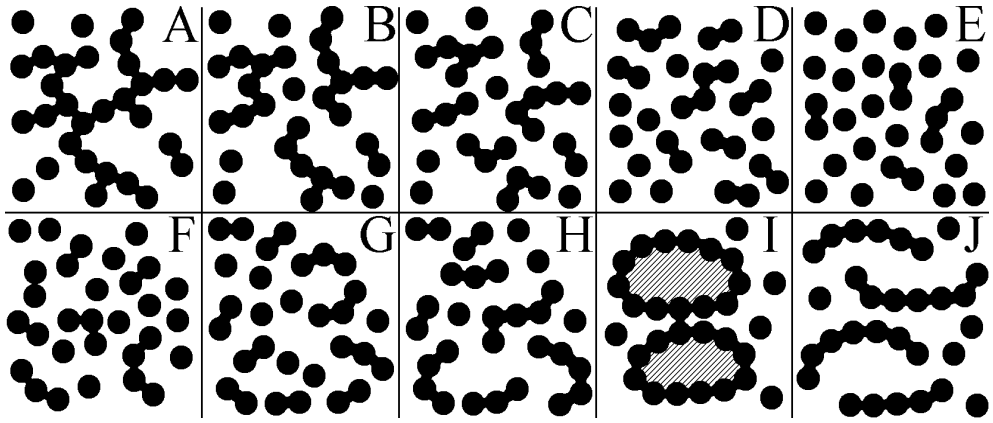


Figure 9.15: A schematic illustration of the change in structure for the flocks of coagulated cement particles. Illustrations **F** to **J** are (most likely) exaggerated in the sense that the number of junctions J_t is too high. However, the idea of spatial reconfiguration applies regardless.

With the above, it could be suggested that the string-of-pearls-structure applies for the first part of the experiment $0 \leq t < 25\,\mathrm{s}$, while the near-spherical-structure

starts to appear in the time period of $25\,\mathrm{s} \leq t \leq 45\,\mathrm{s}$. A (exaggerated) schematic illustration of this idea is shown in Figure 9.15. The alphabetic markings in this figure, corresponds to the markings shown in Figure 9.14. In illustrations **A** to **E** of Figure 9.15, it is shown how the string-of-pearl-structure is broken down into smaller components (occurring at increased rotational frequency f_o). However, when the newly dispersed cement particles (or small flocks of cement particles) start to coagulate again (occurring at reduced rotational frequency f_o), they form a kind of near-spherical-structure as shown with illustrations **F** to **I**. For the latter case, a much higher apparent phase volume is formed than anticipated. This results in a larger shear viscosity η and hence in larger measured torque $\hat{T}$ than anticipated with the theory presented in Section 9.3.

Of course, the above text is just a hypothesis. Nevertheless, it should be clear that it is not unlikely that there exists a difference between the type of structure for the cement flocks at the beginning and at the end of the experiment. The first structure has about 29 minutes to form and do so under perikinetic conditions, while the latter have only tens of seconds to form and do so under orthokinetic conditions.

Attempt to Compensate for Secondary Effect

The above-mentioned difference between the experimental- and the computed torque occurs because the theory of Section 9.3 does not account for any secondary effect. One way to compensate for this effect is by modifying the viscometric values B and $\sqrt{C_\mathrm{y}}$ (see Equations 9.9 and 9.8). The idea is to include the additions μ_se and $\tau_\mathrm{o}^\mathrm{se}$ into these two viscometric values. This is shown with the Equations 9.12 and 9.13. It should be clear that the general form of the shear viscosity Equation 9.7 always remains unchanged, when making this type of maneuver.

$$B = \mu + \tilde{\mu} + \mu_\mathrm{se} \tag{9.12}$$

$$\sqrt{C_\mathrm{y}} \;\; = \tau_\mathrm{o} + \tilde{\tau}_\mathrm{o} + \tau_\mathrm{o}^\mathrm{se} \tag{9.13}$$

Some number of μ_se and $\tau_\mathrm{o}^\mathrm{se}$ functions have been tested to compensate for the **secondary effect**, but without any real success. Some of these are listed below.

$$\left\{\mu_\mathrm{se} = \frac{p\,e^{-(\dot{\gamma}/q)}}{\dot{\gamma}^2+1} \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = 0\right\} \quad \vee \quad \left\{\mu_\mathrm{se} = 0 \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = \frac{p\,e^{-(\dot{\gamma}/q)}}{\dot{\gamma}^2+1}\right\} \tag{9.14}$$

$$\left\{\mu_\mathrm{se} = p\,\frac{\tilde{\Theta}}{\tilde{\Theta}+1} \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = 0\right\} \quad \vee \quad \left\{\mu_\mathrm{se} = 0 \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = p\,\frac{\tilde{\Theta}}{\tilde{\Theta}+1}\right\} \tag{9.15}$$

$$\{\mu_\mathrm{se} = p\,U_3^q \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = 0\} \quad \vee \quad \{\mu_\mathrm{se} = 0 \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = p\,U_3^q\} \tag{9.16}$$

$$\{\mu_\mathrm{se} = m\,\dot{\gamma}^n \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = 0\} \quad \vee \quad \{\mu_\mathrm{se} = 0 \quad \wedge \quad \tau_\mathrm{o}^\mathrm{se} = m\,\dot{\gamma}^n\} \tag{9.17}$$

The most successful modification (however, with only a slight success) consists of using a shear-thickening term in the effective plastic viscosity B. That is, by using $\mu_\mathrm{se} = m\,\dot{\gamma}^n$ and $\tau_\mathrm{o}^\mathrm{se} = 0$, the best result was produced. With this modification, the shear viscosity Equation 9.7 has the following form:

$$\eta = B + \frac{\sqrt{C_\mathrm{y}}}{\dot{\gamma}} = [\mu + \tilde{\mu} + m\,\dot{\gamma}^n] + \frac{\sqrt{C_\mathrm{y}}}{\dot{\gamma}} \tag{9.18}$$

Using a constant value of m in the shear-thickening term $m\,\dot{\gamma}^n$, produces a much larger difference between $\hat{T}$ and $\hat{T}_\mathrm{c}$ than is already present in Figure 9.11. After some trials, using a non constant m term, dependent on shear rate, seemed to give a (slightly) better result. In particular, by using $m(\dot{\gamma}) = p\exp(-[\dot{\gamma}/\dot{\gamma}_\mathrm{cr}]^q)$ a slight improvement over the existing result is produced ($p > 0 \wedge q > 0$). The term $\dot{\gamma}_\mathrm{cr}$ designates a critical shear rate: The idea is that when the shear rate $\dot{\gamma}$ has dropped down to a certain value $\dot{\gamma}_\mathrm{cr}$, the pre-mentioned **secondary effect** starts to appear; i.e. when $\dot{\gamma}$ has dropped down to $\dot{\gamma}_\mathrm{cr}$, sufficient coagulation has occurred to allow the secondary effect to appear in its full strength. Of course, it would have been more natural to use the function of the type $m = m(U_3)$, but surprisingly with the various types of functions tested, such a maneuver resulted in a worse outcome.

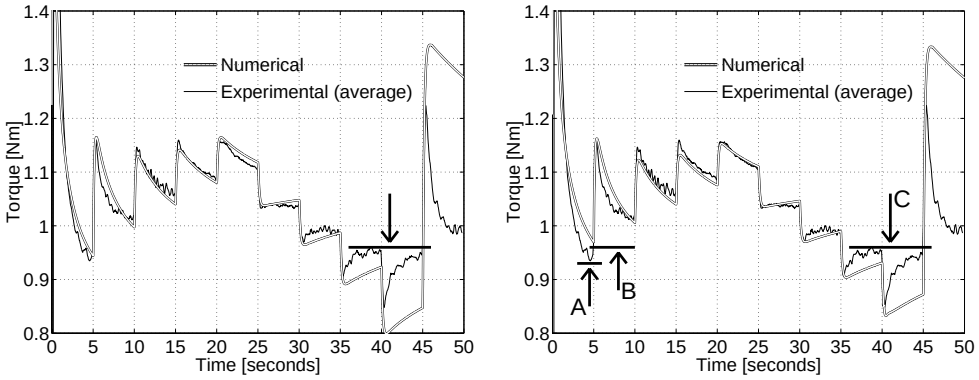


Figure 9.16: Torque $\hat{T}_\mathrm{c}$ generated without (to the left) and with (to the right) a shear-thickening effect $m\,\dot{\gamma}^n$, using $p = 0.8\,\mathrm{Pa}\cdot\mathrm{s}^{n+1}$, $q = 3$, $\dot{\gamma}_\mathrm{cr} = 10\,\mathrm{s}^{-1} \wedge n = 0.6$ (**HMW Ca** $\wedge\, t = 42\,\mathrm{min}$).

A comparison of numerical result where the shear-thickening effect is included, is shown in Figure 9.16. The computed torque $\hat{T}_\mathrm{c}$ generated without a shear-thickening effect is shown in the left illustration, while the right illustration shown a result where such an effect is included. For both illustrations, almost the same viscometric values are used as shown in Table 9.3. The exception is the $a_1\,B_3\,n_3^{2/3}$ term, where it holds the value of $45\,\mathrm{Pa}\cdot\mathrm{s}$ when shear-thickening effect is not included, while it holds the value of $40.5\,\mathrm{Pa}\cdot\mathrm{s}$ when included. With the inclusion of $m(\dot{\gamma})\,\dot{\gamma}^n$, a smaller difference exists between $\hat{T}_\mathrm{c}$ and $\hat{T}$ in the time period between 35 and 45 seconds (the region of arrow C). However, as is shown with the arrows A and B, there is now a larger difference between $\hat{T}_\mathrm{c}$ and $\hat{T}$ in the beginning of the experiment. This last-mentioned difference could be omitted if the shear-thickening term where only active when the condition $d\omega_\mathrm{o}/dt \leq 0$ applies. In taking such an approach, a larger degree of freedom exists in increasing the parameter q without compromising the computed torque in the beginning of the experiment and hence without the need of changing the viscometric values in Table 9.3. However, in taking the liberty of increasing this value without restrictions, some discontinuity in computed torque (and shear rate) starts to appear. In avoiding this, the difference between $\hat{T}$ and $\hat{T}_\mathrm{c}$ always remains.

Lower Coagulation State?

As mentioned previously in this chapter, all viscometric values are determined with the objectives of reproducing the measured torque $\hat{T}$, by numerical means in the best

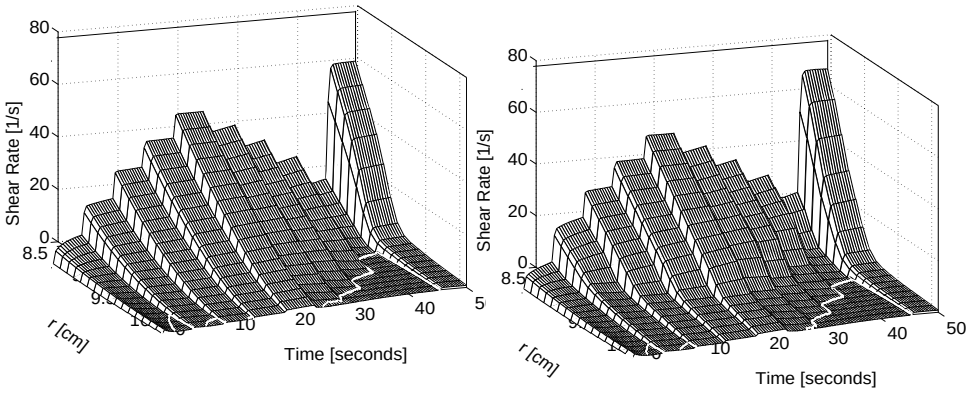


Figure 9.17: Computed shear rate $\dot{\gamma}_c$ generated without (to the left) and with (to the right) a shear-thickening effect $m\,\dot{\gamma}^n$. The same conditions applies here as in Figure 9.16.

fashion. In doing so, the function $K(t)$ is increased by 20 times in the time period between $25\,\mathrm{s} \leq t < 45\,\mathrm{s}$, by using $k_2 = 20\,k_1 = 20\,k_3 = 0.1\,\mathrm{s}^{-3}$ (see Equations 9.10 and 9.11). This was done to reduce the difference between $\hat{T}$ and $\hat{T}_c$ in the time period between 35 and 45 seconds. With the idea about the pre-mentioned secondary effect in mind, this approach might be somewhat incorrect. Figure 9.18 demonstrates the computed results when using $k_2 = k_1 = k_3 = 5 \cdot 10^{-3}\,\mathrm{s}^{-3}$. The same viscometric values are used as presented in Table 9.3. Only the k_2 term is changed, relative to the results presented in Figure 9.11. Also, the shear-thickening term $m\,\dot{\gamma}^n$ is never used in Figure 9.18.

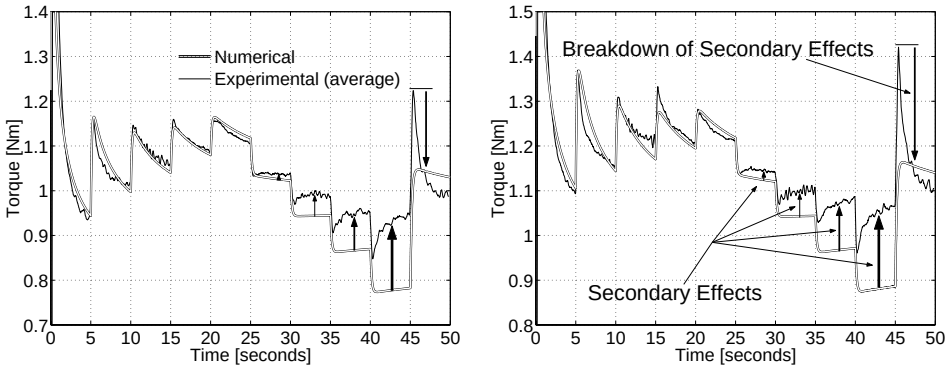


Figure 9.18: Measured- and computed torque $\hat{T} \wedge \hat{T}_c$ at $t = 42\,\mathrm{min}$ (to the left) and at $t = 72\,\mathrm{min}$ (to the right), after the k_2 term is reduced down to $5 \cdot 10^{-3}\,\mathrm{s}^{-3}$ (**HMW Ca**).

As shown with Figure 9.18, by using $k_2 = k_1 = k_3 = 5 \cdot 10^{-3}\,\mathrm{s}^{-3}$, a larger difference between the measured torque $\hat{T}$ and the computed torque $\hat{T}_c$ exists in the period between 25 and 45 seconds. However, in taking this approach the difference is reduced in the period between 45 and 50 seconds. Previously in Figure 9.14, the difference for this time period was marked as "Unknown Effect" because it was inconsistent with the model of secondary effect. But now, in Figure 9.18 the difference is consistent in the sense that the computed torque $\hat{T}_c$ is (almost) always lower than measured torque $\hat{T}$, for the whole time period of $30\,\mathrm{s} \leq t \leq 50\,\mathrm{s}$. The markings

"Breakdown of Secondary Effects" shown in Figure 9.18, corresponds to the steps from **I** to **J** in Figure 9.15.

By reducing k_2 from $0.1\,\mathrm{s}^{-3}$ down to $5 \cdot 10^{-3}\,\mathrm{s}^{-3}$ (the default values are shown in Equation 9.11, Page 213), the coagulation rate function H is decreased in the time period of $25\,\mathrm{s} \leq t < 45\,\mathrm{s}$. This gives a lower reversible coagulation state U_3, as shown in the right illustration of Figure 9.19. This figure demonstrates a comparison of (reversible) coagulation state U_3 with and without the change in k_2 value. When considering how the coagulation is slower in comparison with that of dispersion [67], the severe increase in U_3 shown in the left illustration (between 30 and 45 seconds) could be somewhat overestimated. If so, a value of $k_2 = 0.1\,\mathrm{s}^{-3}$ could be too large.

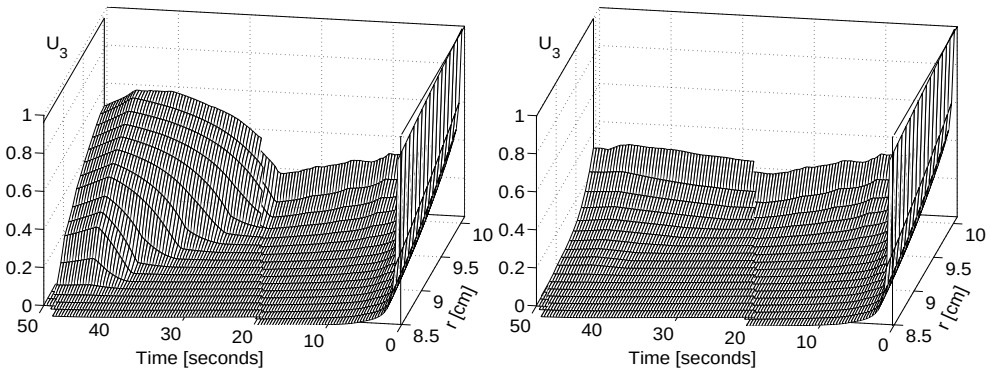


Figure 9.19: The coagulation state U_3 when using $k_2 = 20\,k_1 = 20\,k_3 = 0.1\,\mathrm{s}^{-3}$ (to the left) and when using $k_2 = k_1 = k_3 = 5 \cdot 10^{-3}\,\mathrm{s}^{-3}$ (to the right). (**HMW Ca** $\wedge\ t = 72\,\mathrm{min}$).

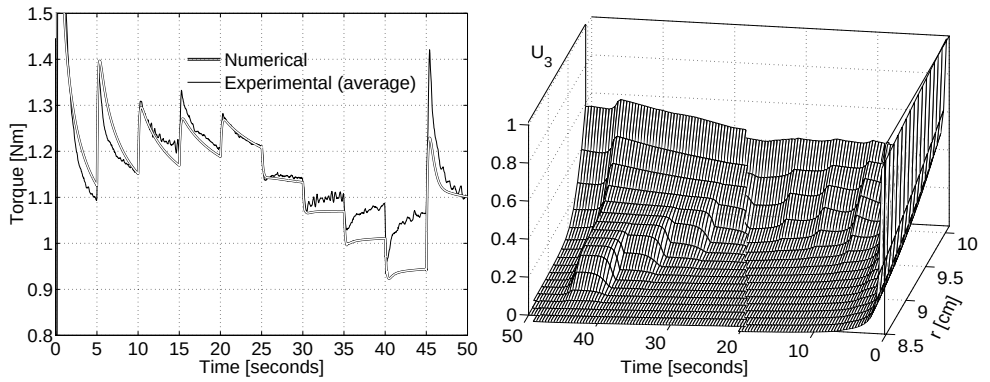


Figure 9.20: Computed torque $\hat{T}_\mathrm{c}$ (to the left) and coagulation state U_3 (to the right), using $m_\mathrm{b} = 0.4\,\mathrm{s}$ and $k_2 = k_1 = 8\,k_3 = 0.4\,\mathrm{s}^{-3}$ (**HMW Ca** $\wedge\ t = 72\,\mathrm{min}$).

There is also another method in reducing the (reversible) coagulation state U_3. Rather than by reducing the value of k_2, a similar effect is gained by reducing the coagulation-memory m_b. In either case of reduced k_2 value (results in lower H) or reduced m_b value (results in lower β), the same effect is inflicted on the memory module $\tilde{\Theta}$, namely a reduction in its value (see Equation 9.4). As shown with Equation 9.5, lower memory module $\tilde{\Theta}$ gives a lower calculated coagulation state U_3. However, a different computed torque profile $\hat{T}_\mathrm{c}$ and a different coagulation state

profile $U_3 = U_3(r,t)$ are produced with the two different approaches of reduced k_2 and m_b. This is shown when comparing the right illustration of Figure 9.19 with the right illustration of Figure 9.20. Both illustrations demonstrate the coagulation state U_3 for the experiment conducted at 72 minutes after water addition. In Figure 9.19, the values of Table 9.3 are used. The same applies for Figure 9.20, except the coagulation-memory m_b is reduced from 15 s down to 0.4 s. Also, the condition of $k_2 = k_1 = 8\,k_3 = 0.4\,\mathrm{s}^{-3}$ is valid in this last mentioned figure; i.e. the values of k_1, k_2 and k_3 are all significantly increased, resulting in a much larger coagulation rate H, than previously used. The shear viscosity Equation 9.18 is used in Figure 9.20, where $m(\dot\gamma) = p\exp(-[\dot\gamma/\dot\gamma_\mathrm{cr}]^q)$ is applied, with $p = 0.8\,\mathrm{Pa}\cdot\mathrm{s}^{n+1}$, $q = 3$, $\dot\gamma_\mathrm{cr} = 10\,\mathrm{s}^{-1}$ and $n = 0.6$.

Summary of Main Results

1. The deviation between the computed torque $\hat{T}_\mathrm{c}$ and the measured torque $\hat{T}$ between 35 and 45 seconds is a result of a (unknown) mechanism, called the secondary effect.

2. The fact that the above-mentioned secondary effect is never present when using the **VHMW Na** polymer (Section 9.4), indicates that this effect is strongly related to the re-coagulation process that occurs in the end of each experiment.

3. Some effort has been made in this section to compensate for this secondary effect, but without any real success.

9.8 SNF

Figure 9.21 demonstrates both the experimental results given by the viscometer and the numerical results given by the software, using the values in Table 9.4. With one exception, the same markings apply in this figure as in Figure 9.3. The exception is in the fourth illustration, marked with "$t = 102\,\mathrm{min}$". There the dotted line represents the result from the second and third batch, while the solid line demonstrates the result from the first batch. The average of these measurements is presented with the dashed dotted line and the double line demonstrates (as always) the numerical result. The reason for this slight change in markings, is that the result produced at 102 minutes after water addition, became quite strange for the second and third batch. As shown, a kind of humpback was formed in the beginning that became impossible to reproduce numerically. Because of this, the numerical simulation consisted of reproducing the measured torque from the first batch only, represented with the solid line.

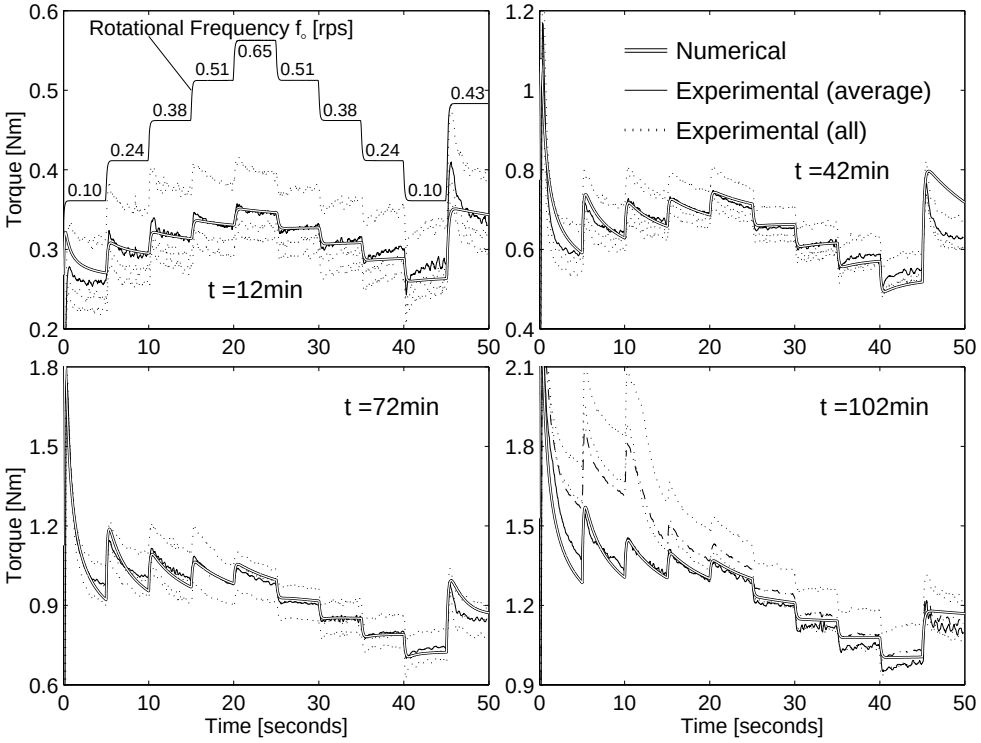


Figure 9.21: Measured- and computed torque $\hat{T} \wedge \hat{T}_c$, for a cement paste using **SNF**.

Some slight difficulties exist in reproducing the measured torque $\hat{T}$ by numerical means for the experiment conducted at 12 minutes after water addition. The low starting torque for this experiment is due to the recent mixing with the Hobart mixer and the good initial dispersing properties of the **SNF** polymer. The difference in the torque $\hat{T}|_{t=4.9\,\mathrm{s}} < \hat{T}|_{t=44.9\,\mathrm{s}}$ is a result of coagulation and (most likely) its byproduct, namely the pre-mentioned secondary effect.

Generally, the secondary effect is much less apparent when using **SNF**, relative to when using **HMW Na** and **HMW Ca**. In particular, for the experiment conducted at 72 minutes after water addition, almost a complete match is gained between the

measured torque $\hat{T}$ and the computed torque $\hat{T}_c$. In generating all illustrations of this figure, the shear-thickening term $m\,\dot{\gamma}^n$ is never used. Also, the k_2 term is kept to its usual value, namely at $k_2 = 20\,k_1 = 20\,k_3 = 0.1\,\mathrm{s}^{-3}$. Other viscometric values used when generating the computed torque $\hat{T}_c$, is shown in Table 9.4.

Table 9.4: Viscometric parameters used in the numerical simulations.

SNF	μ	$a_1\,B_3\,n_3^{2/3}$	τ_o	$a_2\,B_3\,n_3^{2/3}$	U_o	m_a	m_b
	[Pa · s]	[Pa · s]	[Pa]	[Pa]	-	[s]	[s]
$t = 12\,\mathrm{min}$	0.5	30	41	0	$5 \cdot 10^{-2}$	40	10
$t = 42\,\mathrm{min}$	1.0	30	73	30	1	40	4
$t = 72\,\mathrm{min}$	0.5	67.5	110	22.5	1	30	1
$t = 102\,\mathrm{min}$	0.1	75	170	22.5	1	30	0

Figure 9.22 demonstrates the reversible coagulation state U_3 and the computed shear rate $\dot{\gamma}_c$ for the experiment conducted at 72 minutes after water addition. What is immediately apparent when looking at the left illustration, is how low coagulation state U_3 exists although the value of $k_2 = 20\,k_1 = 20\,k_3 = 0.1\,\mathrm{s}^{-3}$ is applied. The reason for this is the low coagulation-memory of $m_b = 1\,\mathrm{s}$ that is applied in the calculation. As was discussed in the last part of Section 9.7, the formation of new junctions J_t due to coagulation, is controlled by the memory module $\tilde{\Theta}$ (see Equations 9.4 and 9.5). This means that the coagulation state U_3 can be reduced either by using a lower coagulation rate H (implemented by using lower k_1, k_2 and k_3 values) or by reducing the value of the memory function β (implemented by using lower m_b value).

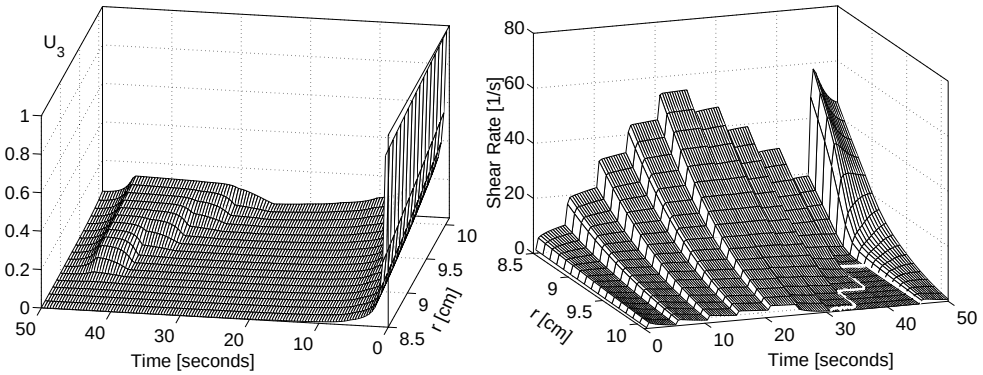


Figure 9.22: The coagulation state U_3 (to the left) and shear rate $\dot{\gamma}$ (to the right) as a function of radius r and time t, when using **SNF** ($t = 72\,\mathrm{min}$).

Summary of Main Results

1. Generally, the secondary effect is much less apparent when using **SNF**, relative to when using **HMW Na** and **HMW Ca**.

2. In particular, for the experiment conducted at 72 minutes after water addition, almost a complete match is gained between the measured torque $\hat{T}$ and the computed torque $\hat{T}_c$.

9.9 Summary and Conclusion

9.9.1 General Findings

In reproducing the measured torque by numerical means, it was necessary to introduce a fading memory into the shear viscosity equation η (Section 9.3.1). The resulting two memory modules used, are defined by Equations 9.3 and 9.4 (Page 210). Also, the introduction of yield value into the shear viscosity η was necessary (Section 9.3.2). Two types of yield values are presented, namely τ_{o} and $\tilde{\tau}_{\mathrm{o}}$. The former ($\tau_{\mathrm{o}}$) is related to the permanent coagulation state $J_{\mathrm{t}}^{\mathrm{p}}$ of the cement particles, while the latter ($\tilde{\tau}_{\mathrm{o}}$) is related to the reversible coagulation state J_{t}. The same type of relationship applies for the plastic viscosity μ and its thixotropic counterpart $\tilde{\mu}$.

The coagulation rate function H used in the simulation, could not be presented with a constant. After some number of tests, a function of $H(\dot{\gamma},t) = K(t)/(\dot{\gamma}^2+l)$ gave the best result (see Equation 9.10, Page 213). This function contains the properties of $dH/d\dot{\gamma} \leq 0 \; \forall \; \dot{\gamma} \geq 0$, something not to be expected, at least by a first consideration (see Section 9.3.3).

9.9.2 Coagulation, Dispersion and Re-Coagulation

The excellent results shown in Section 9.4 (**VHMW Na**) and in Section 9.8 (**SNF**), demonstrate that the theory of Section 9.3 comes a long way in describing the mechanism occurring inside the cement paste. More precisely, this accomplishment gives a good indication of that thixotropy is governed by a combination of reversible coagulation, dispersion and then re-coagulation of the cement particles. This is so, since the computed torque $\hat{T}_{\mathrm{c}}$ does overlap the measured torque $\hat{T}$. However, in Section 9.6 (**HMW Na**) and Section 9.7 (**HMW Ca**), a peculiar deviation between the computed torque $\hat{T}_{\mathrm{c}}$ and the measured torque $\hat{T}$ start to appear in the time period between 35 and 50 seconds.

The fact that the above-mentioned deviation between $\hat{T}_{\mathrm{c}}$ and $\hat{T}$ is never present when using the **VHMW Na** polymer (Section 9.4), indicates that this torque deviation is somehow related to the re-coagulation process that occurs in the end of the experiment. This is so since the only time when there is a good match between the measured torque $\hat{T}$ and the computed torque $\hat{T}_{\mathrm{c}}$, is when re-coagulation is non existent (**VHMW Na**) or is very small (**SNF**).

9.9.3 Shear Rate

With Figures 9.4, 9.9, 9.10 and 9.17, it becomes evident what a great impact the shear viscosity η has on the shear rate $\dot{\gamma}$. With the different rheological values applied, the different shear rates $\dot{\gamma}$ are produced. The same type of conclusion is a fact, when using a pure Bingham fluid in a steady state condition, as shown with Equation 3.28 (Page 59). Also, the rapidly changing boundary R_{s} between the solid state Ω_{e} and the viscoplastic state Ω_{p} is very much dependent on the rapidly changing viscometric values B and $\sqrt{C_{\mathrm{y}}}$ that apply for the cement paste (see Figures 9.12 and 9.13, Page 223). This movement of the solid boundary, provides an additional influence on the shear rate $\dot{\gamma}$. When present, the boundary is shown with a white line and is localized by plotting the function $-\mathrm{II}_{\mathbf{S}}^{\mathrm{P}} = C_{\mathrm{y}}$ (see Figures 9.9, 9.17 and 9.22).

When using a rotating type of viscometer, where the velocity profile can be described with $\mathbf{v} = v_\theta(r,z,t)\,\mathbf{i}_\theta$, the shear rate $\dot{\gamma}$ and the viscoplastic shear viscosity η

must be solved simultaneously through the governing Equation 7.6 (Page 157). This approach is in accordance with the theory of continuum mechanics, and is the only way in calculating the correct shear rate $\dot{\gamma}$, shear viscosity η and hence the correct torque $\hat{T}$. Of course, because of the sheer complexity of the governing equation, it must solved by numerical means, giving the computed counterparts $\dot{\gamma}_{\mathrm{c}}$, $\eta(\dot{\gamma}_{\mathrm{c}})$ and $\hat{T}_{\mathrm{c}}$. Different workers seem not to appreciate the importance of this approach. Some authors give no information about how the shear rate is calculated, giving the impression that it is calculated by analytical means and then as a function of geometry and angular velocity ω_{o} alone. Under steady state condition, and when the test material consists of a Newtonian fluid ($\eta = \text{constant}$), such an approach becomes valid. This is apparent when using $\tau_{\mathrm{o}} = 0$ in Equation 3.28, which results in Equation 9.19

$$\dot{\gamma}_{\mathrm{n}} = \frac{2\,\omega_{\mathrm{o}}}{r^2}\left(\frac{1}{R_{\mathrm{i}}^2} - \frac{1}{R_{\mathrm{o}}^2}\right)^{-1} \qquad [\text{when } \eta = \mu = \text{constant} \wedge \partial v_\theta/\partial t = 0] \qquad (9.19)$$

Figure 9.23 demonstrates a comparison between the shear rate $\dot{\gamma}_{\mathrm{n}}|_{r=R_{\mathrm{i}}}$ calculated by Equation 9.19 (Newtonian fluid under steady state condition), with the shear rate $\dot{\gamma}_{\mathrm{c}}|_{r=R_{\mathrm{i}}}$ extracted by solving Equation 7.6 (viscoplastic fluid under transient condition). The computed shear rate $\dot{\gamma}_{\mathrm{c}}|_{r=R_{\mathrm{i}}}$ is extracted from the **HMW Ca**-case (to the left) and from the **SNF**-case (to the right). Both results apply at $t = 72\,\mathrm{min}$, and the viscometric values used, are shown in Tables 9.3 and 9.4. From this figure it is evident that by using an equation, like Equation 9.19, when calculating the shear viscosity η and torque $\hat{T} = 2\,\pi\,R_{\mathrm{i}}^2\,h\,\tau|_{r=R_{\mathrm{i}}} = 2\,\pi\,R_{\mathrm{i}}^2\,h\,(\eta\,\dot{\gamma})|_{r=R_{\mathrm{i}}}$ of a viscoplastic fluid, an incorrect result will be imminent.

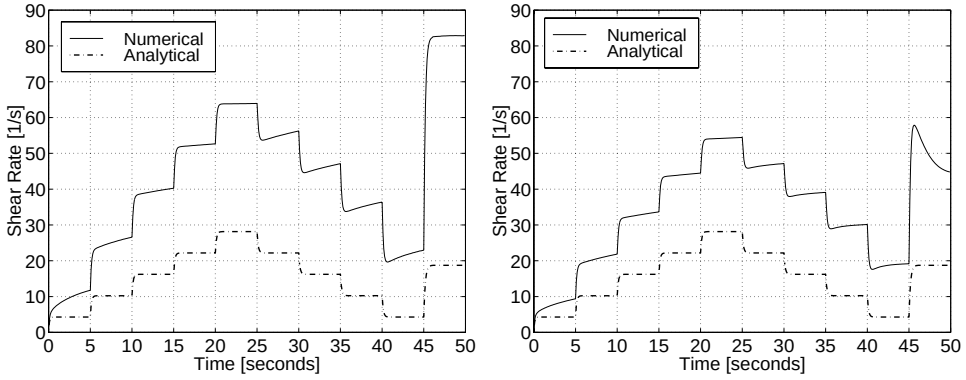


Figure 9.23: Comparison of a "Newtonian" shear rate $\dot{\gamma}_{\mathrm{n}}|_{r=R_{\mathrm{i}}}$ (marked as "Analytical") with computed shear rate $\dot{\gamma}_{\mathrm{c}}|_{r=R_{\mathrm{i}}}$ of viscoplastic fluid (marked as "Numerical"). The former is generated by using Equation 9.19, while the latter by solving Equation 7.6. To the left is the **HMW Ca**-case, while to the right is the **SNF**-case. Both results apply at $t = 72\,\mathrm{min}$.

9.9.4 Viscometric Values

With the overall adequate results of this chapter, the theory of Section 9.3 comes a long way in describing the actual occurrences in the cement particle suspension (i.e. in the cement paste). Therefore, the viscometric values shown in Tables 9.1, 9.2, 9.3 and 9.4 can be considered to represent a good estimation of the rheological response of the cement pastes. These values are plotted in Figure 9.24.

It is interesting to note in Figure 9.24, how small values and little changes apply for the plastic viscosity μ. The same behavior is observed for the mortar experiments, as is shown in Section 6.2. A familiar crossing in yield value τ_o is present between **SNF**-case at the one hand, and the cases of **HMW Na** and **HMW Ca** at the other. This crossing is marked with a box, shown in Figure 9.24. This crossing is also observed for the mortar case, and is marked with the capital letter **A** in Figure 6.1 (Page 104) and in Figure 6.17 (Page 118). Also, roughly the same rearrangement in yield value τ_o exists between the different polymer cases. This overall similarity between the mortar and cement paste is not unexpected, given their relationship as discussed in the beginning of Section 6.4.1.

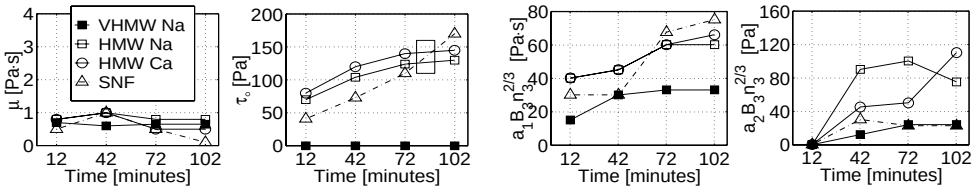


Figure 9.24: Plot of the four primary rheological values μ, τ_o, $\tilde{\mu} = (a_1\, B_3\, n_3^{2/3})\, U_3^{2/3}$ and $\tilde{\tau}_o = (a_2\, B_3\, n_3^{2/3})\, U_3^{2/3}$ (using $U_3 = 1$). Values are extracted from Tables 9.1, 9.2, 9.3 and 9.4.

A plot of the thixotropic terms $\tilde{\tau}_o = (a_2\, B_3\, n_3^{2/3})\, U_3^{2/3}$ and $\tilde{\mu} = (a_1\, B_3\, n_3^{2/3})\, U_3^{2/3}$, using $U_3 = 1$, is made in the two right illustrations of Figure 9.24. The much larger value of $\tilde{\mu}$, relative to μ, is generally not valid near the inner cylinder, because of how the reversible coagulation state $U_3 = J_t/n_3$ becomes quickly very low. This is shown for example in Figures 9.9 and 9.10 (Page 220), and in Figure 9.22 (Page 232).

Although the yield value τ_o in Figure 9.24 is always zero for the **VHMW Na**-case, then this is not true for its thixotropic counterpart $\tilde{\tau}_o$. For self-compacting concrete, it is desirable to keep its yield value τ_o small (to make the concrete self-flowing/self-leveling), and the thixotropic counterpart $\tilde{\tau}_o$ large: During casting of such concrete, then by the recent agitation[9], the $\tilde{\tau}_o$ value is low, meaning that the sum $\tilde{\tau} + \tilde{\tau}_o$ is also low. This makes the concrete self-leveling. Since the formwork is gradually filled with concrete (i.e. not filled all at once), the cement particles of the first placed concrete will have time to re-coagulate, resulting in a relatively quick increase in $\tilde{\tau}_o$. By this, the first layers of concrete will quickly become more self-bearing. With additional layers of concrete, which also quickly re-coagulate, the result will be in an overall lower formwork-pressure[10] (compared to the hydrostatic pressure). This is very important in relation to cost in formwork construction and/or in terms of allowing larger formwork to be filled with concrete during the same day. In this sense, it is beneficial to have a small τ_o value and larger $\tilde{\tau}_o$ value. Such condition applies for the **VHMW Na**-case, but not for the cases of **HMW Na** or **HMW Ca**, and applies least for the **SNF**-case. This gives that the **VHMW Na** polymer is promising for use in self-compacting concrete. Also, increasing $\tilde{\mu}$ and $\tilde{\tau}_o$ values after the cast, will reduce the possibility for segregation of the larger aggregates.

[9] That is, by the recent dispersion of the cement particles, inside the concrete ($J_t \approx 0$).

[10] In terms of mathematics, this means that the condition $(\mathbf{S}^E : \mathbf{S}^E)/2 < \tau_y^2 = (\tau_o + \tilde{\tau}_o)^2$ becomes valid, because of the continuous increase in $\tilde{\tau}_o$ value. This condition inhibits the viscoplastic state in becoming valid and eliminates the concern for the onset of a much larger pressure on the formwork, namely the onset of hydrostatic pressure. The transition from a solid state to a viscoplastic state is described with Equation 3.4 (Page 53).

The strange drop in $\tilde{\tau}_o$ for the **HMW Na**-case at 102 minutes, might rather be related to some experimental error. Because of how the current approach, in retrieving the rheological values in Tables 9.1, 9.2, 9.3 and 9.4, is very time consuming (both in terms of computations and in terms of "tuning in" the correct viscometric values), then no error estimation is made for these values.

It should be clear that the $\tilde{\tau}_o$ and $\tilde{\mu}$ values are not only dependent on the reversible coagulation state J_t. Just as applies for τ_o and μ, they are also (at least) dependent on phase volume Φ and surface roughness. This dependency is then present in the variables $a_1\, B_3$ and $a_2\, B_3$. When creating the (limited) rheological model for the cement paste, presented in Section 2.4.1 (see Equation 2.25, Page 26), it became clear that (at least) the phase volume Φ, surface roughness and coagulation state $J_t^{tot} = J_t^p + J_t$, are the controlling material parameters in changing the value of the overall shear viscosity $\eta = [\mu + \tilde{\mu}] + [\tau_o + \tilde{\tau}_o]/\dot{\gamma}$ as a function of time. A summary of this is given in Section 2.6.3. The exact relationship between J_t and η has been the subject of this chapter.

9.9.5 Recommended Future Research

The good results shown in this chapter demonstrate that the theory of Section 9.3 comes a long way in describing the mechanism occurring inside the cement paste. However, the issue about the secondary effect is not resolved in this thesis. The number of different possibilities in changing the shear viscosity η in the attempt to compensate for this, is virtual infinite. Rather than focusing on the functional form of the shear viscosity η, it is the author's opinion that the best next step is by concentrating on Equation 9.11 (Page 213). In the beginning of this work, it was believed that the change in shear rate $\dot{\gamma}$ would follow the change in the angular velocity ω_o of the outer cylinder (R_o), i.e. if $\dot{\gamma}$ would be increasing, then so would ω_o (and vise versa). When observing animation movies of the numerical results, then this is apparently not so. Often, when the angular velocity ω_o is decreasing, the shear rate $\dot{\gamma}$ is increasing at some locations. The reason for this is related to the movement of the boundary between the viscoplastic state Ω_p and the solid state Ω_e during re-coagulation (i.e. during stiffening) of the test material. As such, the function $K(t)$ (Equation 9.11) should rather depend on $d\dot{\gamma}/dt$ instead of on $d\omega_o/dt$. Implementing such condition and then testing it, is a bit time consuming, and time for such maneuver was not present in the end.

Chapter 10

Particle Migration

10.1 Introduction

Changes in Shear Rate

As is apparent with Equation 3.28 (Page 59), the shear rate $\dot{\gamma}$ in the ConTec viscometers is rapidly changing with location. This is also evident in Figure 10.1, which shows isoplot of the computed shear rate $\dot{\gamma}_c$, calculated from the numerical velocity profile V_k(i,j) of Figure 8.19 (Page 199). As demonstrated in this figure, the shear rate is highest near the inner cylinder and lowest near the outer cylinder.

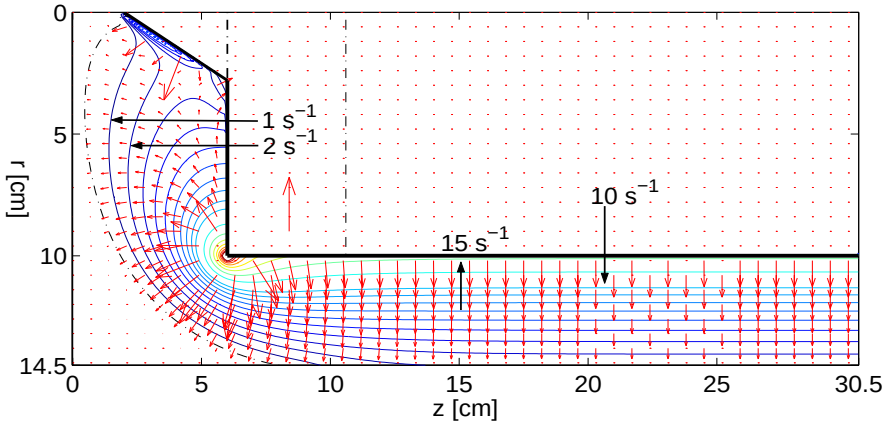


Figure 10.1: Gradient field of shear rate: $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_c$. The length of the vector, shown inside the inner cylinder, is $\|\nabla\dot{\gamma}\| = 10^3\,\mathrm{m}^{-1}\mathrm{s}^{-1}$. The vector concentration in $z \in [20, 25]\,\mathrm{cm}$ is diluted so their actual length can be observed. The isolines are of the shear rate $\dot{\gamma}_c$ and consist of the following values: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12.5, 15, 17.5, 20, 22.5, 25, 27.5, 30]\,\mathrm{s}^{-1}$. The dashed dotted line demonstrates the boundary between a solid state and a viscoplastic state.

Physical Aspects of Particle Migration

The fresh concrete is a (coarse) particle suspension, where it is the gravel particles (here, the $2 - 16\,\mathrm{mm}$ aggregates) that can be modeled as the suspended particles and

with the $0-2\,\mathrm{mm}$ mortar as the surrounding matrix[1]. Migration of the suspended particles from the region of high shear rate, to the region of low shear rate has been reported [9, 70]. The reason for this phenomenon is related to a certain kind of diffuse process, induced by shearing [70]. Here, this phenomenon is rather related to the following: In Section 2.4.1, the (direct and indirect) collision rate $\dot{n}$ between the cement particles is calculated as $\dot{n} = (\tilde{N}_{\mathrm{dc}} + \tilde{N}_{\mathrm{ic}})\,\dot{N} = (\tilde{N}_{\mathrm{dc}} + \tilde{N}_{\mathrm{ic}})\,\dot{\gamma} \propto \dot{\gamma}$ (Page 26); i.e. the collision rate $\dot{n}$ is directly proportional to the shear rate $\dot{\gamma}$. Extrapolate this result to concrete, one could argue that the collision rate $\dot{n}$ between the gravel particles is proportional to the shear rate: $\dot{n} \propto \dot{\gamma}$. If so, the collision rate $\dot{n}$ is also decreasing with increasing radii r (as applies for the shear rate $\dot{\gamma}$). This means that the gravel particles are "pushed" by collisions away from the region of highest collision rate $\dot{n}$, towards the outer cylinder and into the serrated region of inner cylinder. This process is observed as gravel migration, leaving a concrete that is very rich in mortar, behind in the region near the inner cylinder. This is shown in Figures 10.2 and 10.3. Also, for the gravel particles to be able to migrate towards the outer cylinder and into the serrated region of inner cylinder, the mortar that resides there, between the gravel particles, has to be consequently squeezed toward the region near the inner cylinder.

Figure 10.2: To the left: In some rear occasions, particle migration inside the ConTec Viscometer 4 occurred. The thickness of the migrated layer shown, is measured to be around 5 mm. Center: Gravel migration inside the ConTec **BML** Viscometer 3, for the stiffer type of concrete batches of this thesis. To the right: Schematic figure of gravel distribution $\Phi = \Phi(r, z)$ after a viscometric measurement (the white tube shown, will be discussed shortly).

Another physical phenomenon could be present at the same time, also responsible for the particle (or gravel) migration. This is the effect of dilatancy and is slightly mentioned in Section 3.2.2 (see Figure 3.3, Page 54). In a suspension of densely packed suspended particles (i.e. large phase volume Φ), like what is present here, the gravel distribution Φ must change to permit the gravel particles to flow past one another. This means a withdrawal (or rather suction) of matrix, from the region of smallest deformation (or smallest $\dot{\gamma}$) into region of largest deformation (or largest $\dot{\gamma}$) and hence, change the gravel concentration in the process. This phenomenon would continue until a sufficient amount of matrix exists between the gravel particles, in the region near the inner cylinder. More precisely, this means that the matrix near the

[1] In this chapter, the **matrix** is defined as all particles below 2 mm in diameter. This means the $0-2\,\mathrm{mm}$ aggregates, cement particles, water and other solid particles in the same size range, defines the matrix. Because there is no distinct physical boundary between the suspended particles and the matrix, it is difficult to define the "true" matrix. With this definition, the phase volume Φ [9] is then equal to $\Phi = \delta V_{2-16}/\delta V$ where δV_{2-16} is the amount of volume occupied by gravel particles inside the continuum particle (CP). The term δV has been defined previously and represents the total volume of the CP. The phase volume of cement paste is discussed in Footnote 33, Page 48.

outer cylinder and in the serrated region of inner cylinder are sucked into the area of the highest shear rate, namely in the direction of $\nabla\dot{\gamma}$. With a (observed) constant volume of the overall suspension, the gravel particles must be consequently pushed in the opposite direction, namely in the direction of $-\nabla\dot{\gamma}$. A interesting experiment has been made by Mork [78] in his doctoral thesis, in measuring this phenomenon. Basically it consist of submerging a tube filled with liquid, into a concrete sample. The tube configuration is shown in the right illustration of Figure 10.2: The tube is in a vertical position, placed near the inner cylinder. Both ends of the tube are open. On measurement, the liquid inside the tube is sucked into the zone near the inner cylinder. In repeating the same experiment right afterwards, when the concrete batch is in a migrated state, the liquid remained in the tube.

A third physical phenomenon could also be accountable for the particle migration. This will be called the "confinement effect", and becomes apparent at low ratio of $D_{\text{flow}}/D_{\max}$. This effect will be discussed in Section 10.3.1. The mechanism for the collision, dilatancy and the confinement effect are summarized in Section 10.3.8.

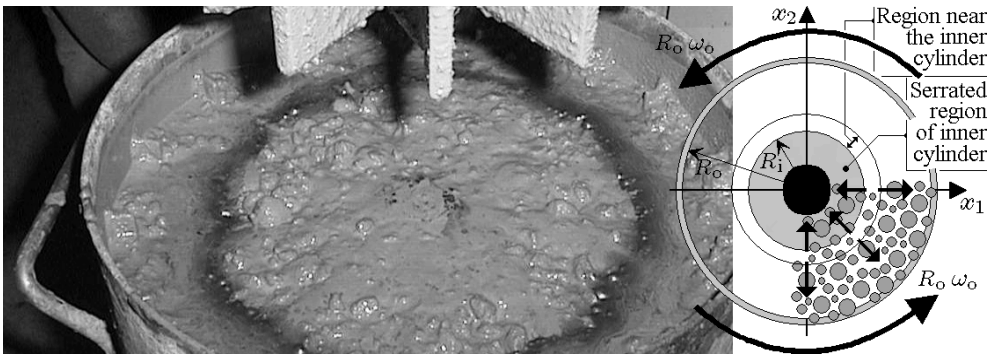


Figure 10.3: An extreme case of gravel migration inside the ConTec **BML** Viscometer 3, for a very fluid fresh concrete (this batch is however not of this thesis). The gravel particles have migrated from the zone of high shear rate, towards the zone of low shear rate.

In this work, the change in gravel concentration, from $\Phi = \text{constant}$ to $\Phi = \Phi(r, z)$, during a measurement, will be often referred to as **gravel migration**, but generally in the literature, such process is usually designated as **particle migration** [9, 70]. In mathematical terms, this experience can be stated with $\mathbf{q} = -\vartheta\,\nabla\dot{\gamma}$, where the migration coefficient ϑ describes how much amount of gravel particles $\mathbf{q}$ should migrate for the given difference in shear rate $\nabla\dot{\gamma}$ (this principle is similar to the Fourier law of heat flux: $\mathbf{q} = -k\,\nabla T$ given in Appendix B.7.2). Using this approach, it is apparent that the direction and the potential magnitude of gravel migration is related to the gradient of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}$, plotted as arrows in Figure 10.1.

Experimental Observation

The objective of Chapter 10, is not to prove that gravel migration occurs. Rather, the objective here is to characterize and analyze how much this phenomenon influences the viscometric values retrieved. In this thesis, a "proof" of gravel migration is never made in a systematic sense, by direct measurements. However, this phenomenon was always observed after a viscometric measurement on the concrete batches of this thesis. This was done both with a direct observation like presented in Figure 10.2 and especially when emptying the test material back to the Gustav Eirich mixer.

For the latter case, gravel migration became quite apparent with the difficulty in cleaning the area near the outer cylinder ($r = R_o$) and near the bottom part of the bucket ($z = 0$), due to the high gravel concentration Φ there (the bucket is shown in the center illustration of Figure 3.5 on Page 56). Regardless of the above visual observations, it would have enhanced this thesis if a more systematic analysis on the gravel concentration would have been made. For example, letting the fresh concrete harden to some extent after a viscometric measurement and then taking a cross section of it, to measure the gravel concentration as a function of location: $\Phi = \Phi(r, z)$. This would have given a clearer picture of the extent of gravel migration.

Particle migration inside the ConTec Viscometer 4, when measuring the $0 - 2\,\text{mm}$ mortar, is almost never observed in this work. But like what is shown in the left illustration of Figure 10.2, in some rear occasions this phenomenon occurred. The thickness of the migrated layer shown, is measured to be around 5 mm. This phenomenon is mostly observed when using the fly ash cement at 38°C degrees temperature. There the ratio of yield value to plastic viscosity started to reach around and above $\tau_o/\mu \approx 200\,\text{s}^{-1}$. As will be discussed in Section 10.3.1, the higher the ratio τ_o/μ becomes, the larger the gradient in shear rate $-\nabla\dot{\gamma}$ is formed and hence a larger possibility for particle migration $\mathbf{q} = -\vartheta\nabla\dot{\gamma}$.

None of the concrete batches in this project segregated under manual agitation, since they were designed in such manner. Therefore, gravel migration could not be caused by centrifugal force in the viscometer: Since the concrete could withstand a manual agitation without segregating, it could also (at least) tolerate an acceleration of $g = 9.81\text{m/s}^2$. With $f_{\max} = 0.5$ rps, the maximum centripetal acceleration near the inner cylinder is $R_i\,\omega_o^2 = 0.1\text{m}\,(2\pi\,0.5\text{rps})^2 = 0.99\text{m/s}^2$. This simple calculation demonstrates that the gravitational acceleration is about ten times larger than the centripetal acceleration and therefore it can be concluded that the latter acceleration cannot be responsible for the horizontal segregation observed.

Particle Migration Inside Other Types of Viscometers

Gravel migration is by no means something that applies only to the ConTec **BML** Viscometer 3. For example, this potential is also present inside the BTRHEOM viscometer (a description of this viscometer is given in [50, 51, 27]). Since such process would occur well within the test material (see Figure 10.24), visual observations like presented in Figures 10.2 and 10.3 are hard to gain for the BTRHEOM viscometer.

Objectives

The first objective (Section 10.2) of this chapter, is to quantify the effect of gravel migration by observing the rotational frequency f, 2 cm away from the inner cylinder R_i. In doing this type of investigation, one can gain some idea about how gravel migration influences the true (and unknown) rheological values τ_o and μ of the concrete, in a homogeneous state. This investigation is only applied to the ConTec **BML** Viscometer 3, since particle migration inside the ConTec Viscometer 4, is almost never observed.

The second objective (Section 10.3) is to explore the possibility for a more suitable viscometric geometry to the ConTec **BML** Viscometer 3, in minimizing the probability of gravel migration. This analysis will be based on the gradient in shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}$, where the geometry that produces the lowest difference in the shear rate, i.e. has the smallest vectors (see Figure 10.1), is considered to be the best solution.

10.2 Measuring the Effect of Particle Migration

The objective of this section is to quantify the effect of gravel migration, by measuring the rotational frequency, 2 cm away from the inner cylinder, inside the ConTec **BML** Viscometer 3. To do this, some modifications of the viscometer are made as shown in Figures 10.4 and 10.5. The point is that while gravel migrations is occurring, the rotational frequency f (and hence velocity $v_\theta\,\mathbf{i}_\theta$) should be additionally increasing from what is given by Equation 3.23 (Page 59), due to the plug that is forming. The plug state is now more related to the concentration increase in gravel content Φ near the outer cylinder.

The mechanism for the onset of plug, discussed in Section 3.5 applies only for a homogenous test sample (Φ = constant), where for example, it is the shear rate $\dot{\gamma}$ that marks the onset of plug, then with $\dot{\gamma} = 0$. However, in this chapter, it is also the change in gravel concentration Φ that can mark the onset of plug. When this concentration has increased to a certain value, a rigid body motion will result.

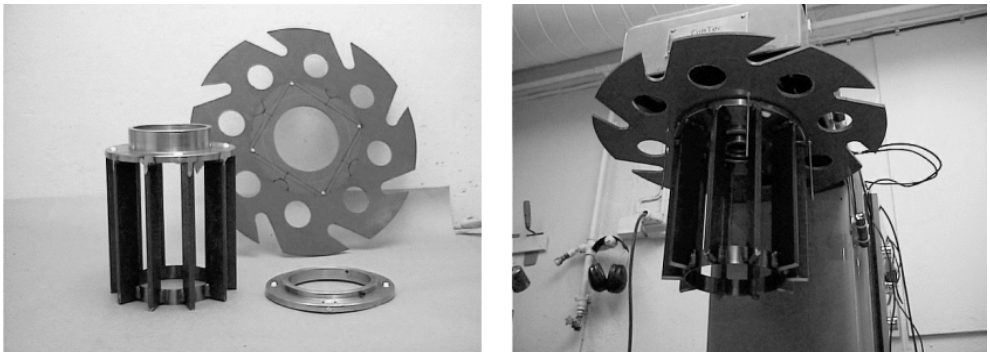


Figure 10.4: To the left: The three basic elements of the velocity measuring device. To the right: After mounting this device on the viscometer, total of four needles are aligned in a vertical position. This is also shown in the right illustration of Figure 10.5.

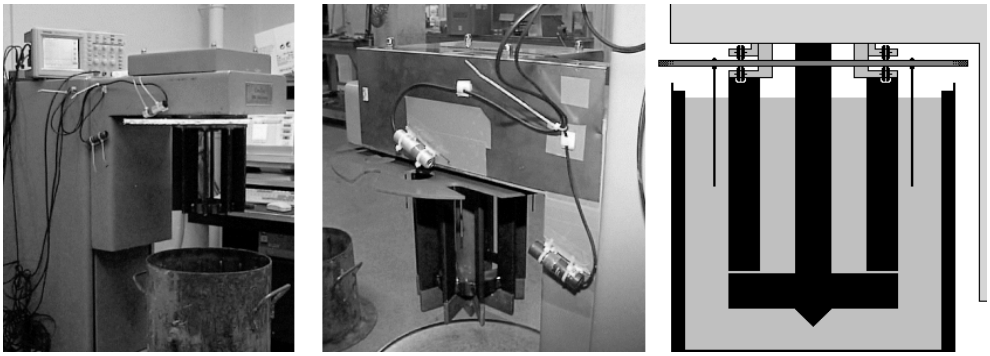


Figure 10.5: The photocell system during developing stage (to the left) and in the final stage (center). To the right is a schematic illustration of the velocity measuring device.

As Figures 10.4 and 10.5 demonstrate, the velocity measuring device basically consists of three objects: Four needles that are submerged into the concrete sample, a freely rotating plastic disc to which the needles are attached, and finally, a photocell

that records the rotational frequency f_{disc} of the plastic disc. The mix design of the concrete batches used here, is as described in Section 4.3.1. The mixing and measuring procedure is as described in Section 5.2.1. The parameter setup used for the viscometer is as shown with Table 3.1 (Page 64).

The results that now follow are of six different concrete batches collected in two groups, marked with the type of lignosulfonates used. The two groups are named "**HMW Na & LMW Na**" and "**VHMW Na**".

10.2.1 HMW Na & LMW Na

Measured Case: Figure 10.6 shows a result from rheological measurements made on fresh concrete, using 0.6% sbwc of **HMW Na** at $w/c = 0.4$. The left illustration demonstrates the measured torque $\hat{T}$ as a function of time. Also shown, is the rotational frequency f_{o} applied for the given time period. The right illustration demonstrates the percentage difference between measured rotational frequency f_{disc} of the test sample, at $r = R_{\text{i}} + 2\,\text{cm}$, and the rotational frequency of a solid f_{o}. The difference is calculated as $100 \cdot (f_{\text{o}} - f_{\text{disc}})/f_{\text{o}}$ and is marked as "Measured:".

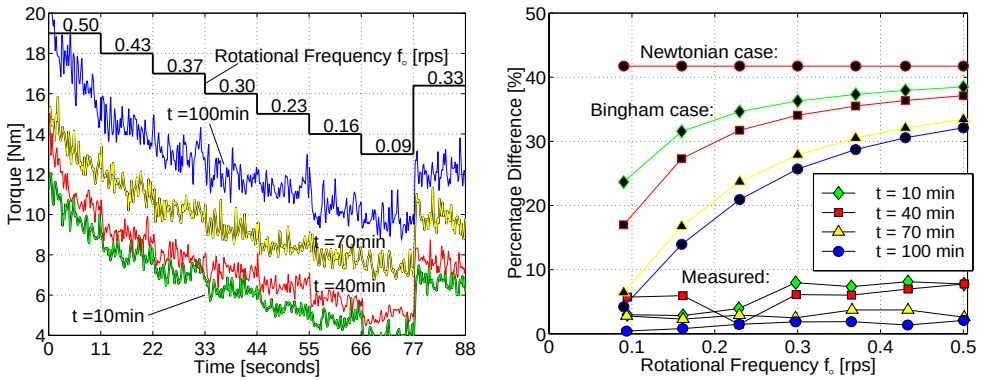


Figure 10.6: Test sample: Fresh concrete, using 0.6% sbwc of **HMW Na** at $w/c = 0.4$. To the left is the measured torque $\hat{T}$ and to the right is the percentage difference between rotational frequency at $r = R_{\text{i}} + 2\,\text{cm}$ and the rotational frequency of a solid f_{o} (i.e. of a rigid body).

Bingham Case: Assuming that no gravel migration is occurring (Φ = constant $\forall\ t$) and therefore using all torque data available when making the data processing (described in Section 3.4.2), the yield values produced are $\tau_{\text{o}}^{\square} = 123, 177, 305$, and 415 Pa. Likewise, the plastic viscosities produced consist of $\mu^{\square} = 41.3, 41.9, 40.0$, and $46.9\,\text{Pa}\cdot\text{s}$. As mentioned in Section 5.2.1, these values apply at 10, 40, 70, and 100 minutes after water addition. Through[2] Equations 3.23 and 3.24 (Page 59), these viscometric values are then used in calculating the rotational frequency $f(R_{\text{i}} + 2\,\text{cm})$ that should be measured if these values were correct. Prior to this type of calculation, Equation 3.32 (Page 66) is always applied in order to calculate the solid boundary R_{s}, whenever (theoretical) plug is occurring. This result is used in Equation 3.24. The rotational frequency that is calculated with the above approach, is designated here with f_{b}. The difference $100 \cdot (f_{\text{o}} - f_{\text{b}})/f_{\text{o}}$ is shown in Figure 10.6 and is marked as "Bingham case:". This is the difference that should be measured if the values $\tau_{\text{o}}^{\square}$

[2]The relationship between v_θ and f is $\boldsymbol{\omega} = 2\,\pi\,f\,\mathbf{i}_{\text{z}} = \mathbf{i}_r \times v_\theta(r,t)\,\mathbf{i}_\theta/r = v_\theta(r,t)/r\ \mathbf{i}_{\text{z}}$.

$\wedge\ \mu^{\square}$ represented the correct (and unknown) viscometric values $\tau_{\mathrm{o}} \wedge \mu$ of the fresh concrete, in a homogenous state.

Newtonian Case: For the values marked with "Newtonian case:", the yield value τ_{o} is set equal to zero, which produces a velocity profile of v_θ independent on viscometric values: $f_{\mathrm{n}} = \omega/(2\,\pi) = v_\theta/(2\,\pi\,r) = f_{\mathrm{o}}\,(1-R_{\mathrm{i}}^2/r^2)/(1-R_{\mathrm{i}}^2/R_{\mathrm{o}}^2)$. The difference $100\cdot(f_{\mathrm{o}}-f_{\mathrm{n}})/f_{\mathrm{o}}$ is equal to $100\cdot[1-(1-R_{\mathrm{i}}^2/(R_{\mathrm{i}}+2\,\mathrm{cm})^2)/(1-R_{\mathrm{i}}^2/R_{\mathrm{o}}^2)] = 42\%$.

As pointed out previously, in the process of gravel migration, the outer region of the test sample (i.e. outside the circle of $r = R_{\mathrm{i}} + 2\,\mathrm{cm}$) becomes more and more packed with gravel particles and therefore resembles more and more a solid. In other words, a plug state (i.e. rigid body motion) is achieved at the outer region due to the accumulations of gravel particles there. The result of such process is that the test material there starts to approach the same rotational frequency as the outer cylinder, namely approaching f_{o}. As is shown in Figure 10.6, this phenomenon is indeed observed: The percentage difference between f_{disc} and f_{o} is below 10% for all cases. This is a much lower difference then should be measured as is shown either with the "Bingham case:" or with the "Newtonian case:".

As shown in Figure 10.6, the characteristics of the "Bingham case:", is that the difference $100\cdot(f_{\mathrm{o}}-f_{\mathrm{b}})/f_{\mathrm{o}}$ approaches the "Newtonian case:" with larger rotational frequency f_{o}. This is to be expected, since with higher rotational frequency f_{o}, the shear rate $\dot\gamma$ increases, leading to $\eta = \mu + \tau_{\mathrm{o}}/\dot\gamma \approx \mu$. The same effect is produced with decreasing yield value τ_{o}.

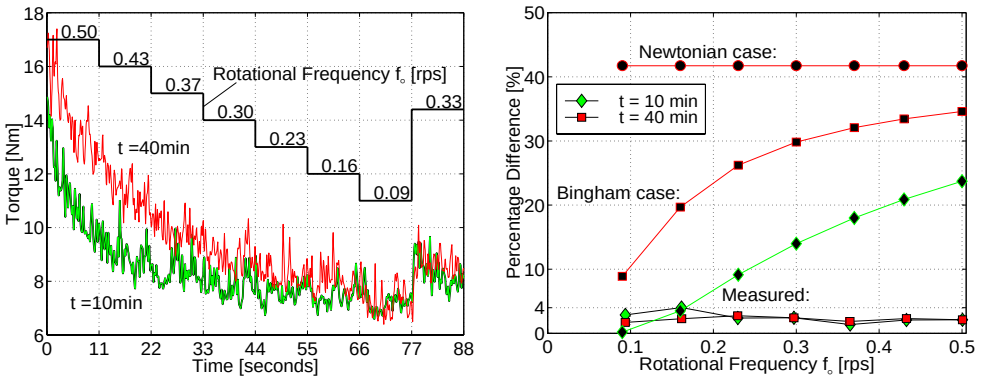


Figure 10.7: Test sample: Fresh concrete, using 0.3% sbwc of **LMW Na** at $w/c = 0.5$. To the left is the measured torque $\hat{T}$ and to the right is the percentage difference between rotational frequency at $r = R_{\mathrm{i}} + 2\,\mathrm{cm}$ and the rotational frequency of a solid f_{o}.

Figure 10.7 shows a result from rheological measurement made on fresh concrete, using 0.3% sbwc of **LMW Na** at $w/c = 0.5$. The left illustration demonstrates the raw torque data $\hat{T}$, while the right illustration demonstrates the percentage difference between measured rotational frequency f_{disc} of the test sample at $r = R_{\mathrm{i}} + 2\,\mathrm{cm}$ and the rotational frequency of a solid f_{o}. The same remarks applies in Figure 10.7 as for Figure 10.6. The viscometric values produced here are $\tau_{\mathrm{o}}^{\square} = [323, 316]\,\mathrm{Pa}$ and $\mu^{\square} = [19.6, 48.2]\,\mathrm{Pa}\cdot\mathrm{s}$.

The one thing that is distinct with the left illustration of Figure 10.7 is the continuous decrease in torque $\hat{T}$, even at the given constant rotational frequency f_{o}. One cannot argue that this decrease is due to thixotropic behavior alone (of the cement paste inside the concrete). Rather, this "breakdown" is (also) related to gravel

migration (this perception is in an agreement with earlier work [129]). This is apparent with the result from Figure 10.8, which demonstrates the measured torque $\hat{T}$ for the corresponding mortar mix, designed from the concrete batch (according to Section 4.3.2). The left illustration demonstrates two measurements made with the mixing and measuring procedure of Section 5.3.3, while the right illustration with the mixing and measuring procedure of Section 5.3.1. The latter mixing and measuring procedure is nominally identical to what applies for concrete. As shown with either illustration in Figure 10.8, no apparent thixotropic breakdown exists for the corresponding mortar. If anything, there is rather a slow thixotropic rebuild occurring, by the re-coagulation process. Since, the mortar inside the concrete, is under similar (or larger) agitation as the mortar in Figure 10.8 (see Section 4.4.1) and since there is no apparent thixotropic breakdown shown in Figure 10.8, then it can be argued that no (or only small) thixotropic breakdown should be present for the corresponding concrete batch in Figure 10.7.

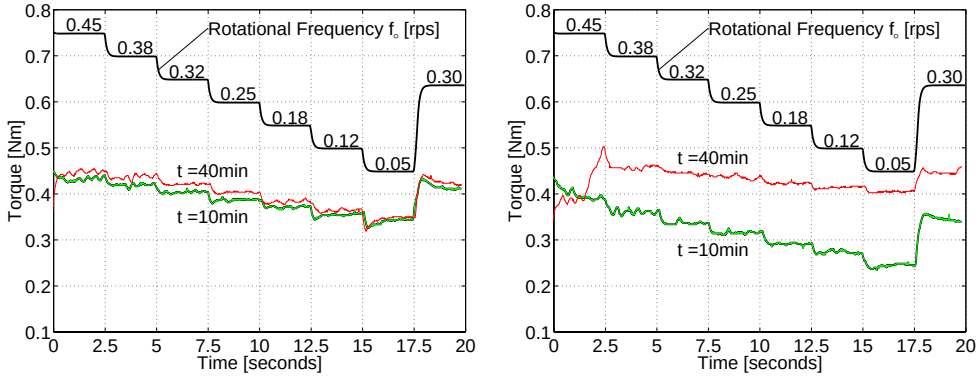


Figure 10.8: Test sample: Mortar, using **LMW Na** at $w/c = 0.5$. Mixing and measuring procedure used are of Section 5.3.3 (to the left) and of Section 5.3.1 (to the right).

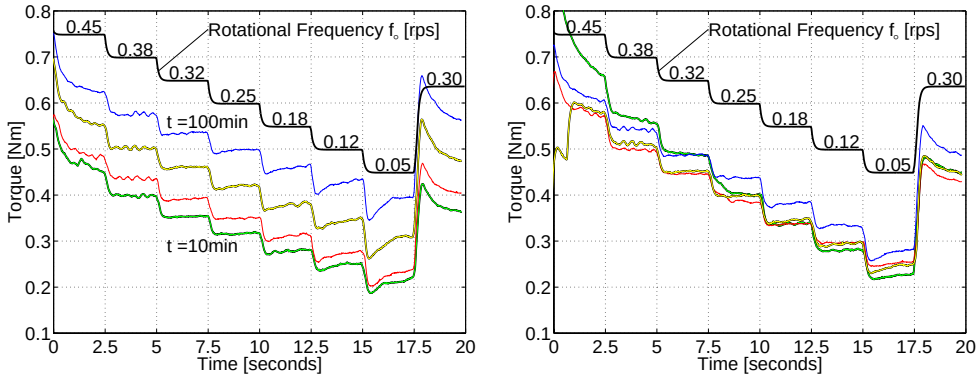


Figure 10.9: Test sample: Mortar, using **HMW Na** at $w/c = 0.4$. Mixing and measuring procedure used are of Section 5.3.3 (to the left) and of Section 5.3.1 (to the right). For both illustrations, the average of 4 batches is used in generating each line.

The same arguments apply for the concrete batch in Figure 10.6. This is apparent with the result shown in Figure 10.9, which demonstrates the measured torque for the

corresponding mortar mix. There, the thixotropic breakdown is finished within 2.5 seconds for the mortar, while the continuous reduction in torque, for the corresponding concrete batch, goes on for at least 22 seconds. The same type of result applies for the concrete batches in Figures 10.11, 10.13 and 10.14, however only based on the mixing and measuring procedure of Section 5.3.3.

It should also be noted that prior to any torque logging, the fresh concrete is pre-rotated for about 10 seconds (see Table 3.1). This means that at the first torque logging (at "$t = 0$ seconds" in the left illustration of Figure 10.7), a great deal of thixotropic breakdown should be finished.

The right illustration of Figure 10.7, demonstrates that the difference between the rotational frequency of the test material f_{disc} and of a solid f_{o} (i.e. of the outer cylinder) is below only 4% at all times, indicating that gravel saturation Φ outside the circle of $r = R_{\text{i}} + 2\,\text{cm}$ occurs apparently very fast. However, as mentioned previously, prior to any torque logging, the fresh concrete is pre-rotated for about 10 seconds. This (default) setup is unfortunate because important information about the torque is missing in this period, namely when the largest changes is gravel concentration Φ is occurring. Hence, if all torque information were available, larger difference than 4% could be anticipated at the beginning (i.e. from $t = -10$ seconds to $t = 0$ seconds).

Since the difference between f_{disc} and f_{o} is always below 4% in Figure 10.7, the reduction in torque $\hat{T}$ shown in the left illustration is most likely due to the continuous gravel migration within the circle of $r = R_{\text{i}} + 2\,\text{cm}$. The possibility for that this reduction is also related to thixotropic breakdown of the remaining fat concrete within the circle, is not completely discarded. However, as demonstrated with Figures 10.8 and 10.9 thixotropic behavior is not expected to be the dominating effect. Thixotropic breakdown could rather be related to the reduction in torque, shown between 77 and 88 seconds in Figure 10.7, since it is expected that gravel migration is long finished before that time period.

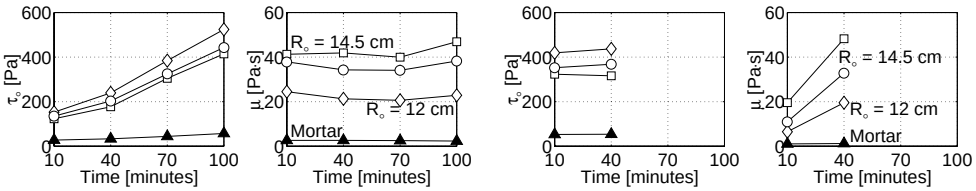


Figure 10.10: Results from the different ways in calculating the viscometric values τ_{o} and μ of the fresh concrete: $\tau_{\text{o}}^{\square} \wedge \mu^{\square}$; $\tau_{\text{o}}^{\bigcirc} \wedge \mu^{\bigcirc}$; $\tau_{\text{o}}^{\diamondsuit} \wedge \mu^{\diamondsuit}$. The two illustrations to the left consist of **HMW Na** at $w/c = 0.4$, while the two to the right consist of **LMW Na** at $w/c = 0.5$.

With the information presented here, it is possible to calculate the viscometric values of the remaining fat concrete sample within the $r = R_{\text{i}} + 2\,\text{cm}$ circle. First, only the torque values between 22 and 77 seconds are used, since the majority of migration seems to be finished after 22 seconds. Plotting the remaining five torque points versus f_{disc}, and otherwise using the same data processing method of Section 3.4.2, gives the results shown in Figure 10.10, marked with a diamond $\diamondsuit$ ($\Rightarrow \tau_{\text{o}}^{\diamondsuit} \wedge \mu^{\diamondsuit}$) and designated as "$R_{\text{o}} = 12\,\text{cm}$". This designation is used, since outer radius of $R_{\text{o}} = 12\,\text{cm}$ must be used, instead of $R_{\text{o}} = 14.5\,\text{cm}$. This is so since f_{disc} is now applied instead of f_{o}.

With the results designated as "$R_{\text{o}} = 14.5\,\text{cm}$" in Figure 10.10, two different approaches are used when calculating viscometric values, marked with a box $\square$ ($\Rightarrow \tau_{\text{o}}^{\square} \wedge \mu^{\square}$) and a circle $\bigcirc$ ($\Rightarrow \tau_{\text{o}}^{\bigcirc} \wedge \mu^{\bigcirc}$). Their calculations are explained in Section 3.4.2.

The "Mortar" measurement is marked with a triangle △ and designates the corresponding mortar mix, using the mixing and measuring procedure of Section 5.3.3.

Figure 10.10 demonstrates that the viscometric values of the mortar △ is much lower than of the remaining fat concrete sample ♢ within the circle of $r = R_i + 2\,\text{cm}$. This indicates that this fat concrete is far from being completely depleted of gravel particles. This is also apparent with the fact that the fluctuating torque (i.e. the amplitude of the random oscillations in torque) are not diminishing with time as shown for example in Figure 10.7. The fluctuation in torque is most likely generated by continuous formation of gravel-bridges as described in Section 3.4.2.

The yield value $\tau_o^{\diamondsuit}$ (or the shear viscosity $\eta^{\diamondsuit}$) of the fat concrete within the circle of $r = R_i + 2\,\text{cm}$ is in reality lower than the (unknown) yield value of the concrete in a homogeneous state[3]. However, as is shown in Figure 10.10, this is not measured; i.e. the yield values designated as "$R_o = 14.5\,\text{cm}$" and marked with ○ and □ (which are supposed to produced the viscometric values for the concrete in a homogenous state) are always measured with a lower yield value τ_o, than measured for the fat concrete ♢ within the circle. Because of this, it is apparent that the values of "$R_o = 14.5\,\text{cm}$" are lower than the true and the desired yield value τ_o of a homogenous fresh concrete.

10.2.2 VHMW Na

Figure 10.11 shows a result of rheological measurements conducted on fresh concrete, using 0.6% sbwc of **VHMW Na** at $w/c = 0.4$. At 10 minutes after water addition, this batch had the characteristics of a self-compacting concrete with the slump of 265 mm and slump flow of 620 mm. Applying the usual data processing for this measurement, produces a negative yield value of $\tau_o^{\square} = -11.8\,\text{Pa}$ (the resulting torque $\hat{T}$ as a function of f_o is shown in Figure 10.18). Overall, the viscometric values produced are $\tau_o^{\square} = [-11.8, 12.4, 55.4, 114]\,\text{Pa}$ and $\mu^{\square} = [36.7, 37.3, 38.7, 43.3]\,\text{Pa}\cdot\text{s}$. The same remarks applies in Figure 10.11 as for Figure 10.6.

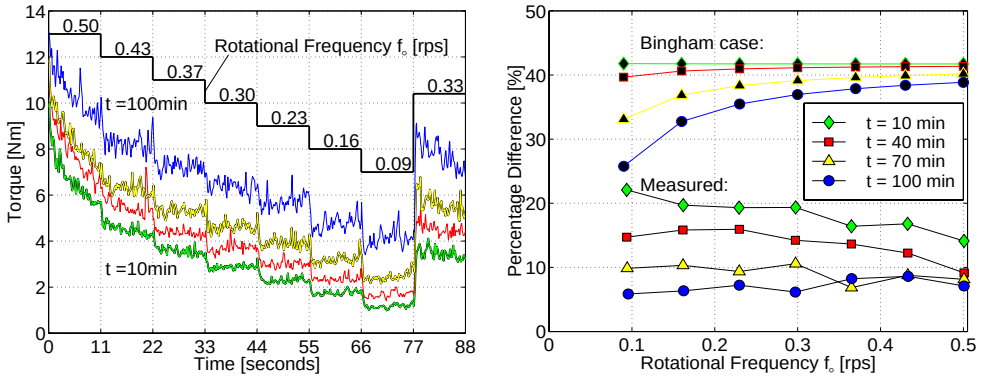


Figure 10.11: Test sample: Fresh concrete, using 0.6% sbwc of **VHMW Na** at $w/c = 0.4$. To the left is the measured torque $\hat{T}$ and to the right is the percentage difference between rotational frequency at $r = R_i + 2\,\text{cm}$ and the rotational frequency of a solid f_o (i.e. of a rigid body). No thixotropic rebuild or breakdown where observed for the corresponding mortar mix.

[3]This is so, since the homogeneous concrete has a higher gravel consternation Φ in a shearing zone. This results in a larger momentum exchange between the gravel particles for this concrete, which again gives a larger shear viscosity η. A summary is given in Section 2.6.3, about the relationship between the momentum exchange and the shear viscosity η, for cement paste.

When calculating the term f_b (Bingham), for the case of 10 minutes after water addition, the yield value of zero is used rather than of $-11.8\,\text{Pa}$. This results in that f_b becomes equal to f_n (Newtonian) for that particular case. As shown in the right illustration of Figure 10.11, then at 10 and 40 minutes after water addition, the difference between f_{disc} and f_o is much higher than measured previously, indicating that plug near the location of $r = R_i + 2\,\text{cm}$ is not occurring. Nevertheless, these two measurements were somewhat inflected by gravel migration. This was observed visually after the rheological test and especially when emptying the bucket (i.e. the outer cylinder) of this batch into the GUSTAV EIRICH mixer, where (like always) one could feel how the gravel concentration was higher at the sides and at the bottom of the bucket, as predicted by Figure 10.1. However, for the first two measurements at 10 and 40 minutes, gravel migration were not as visually apparent as for the other concrete measurements conducted in this thesis. The reason for this is related to the small τ_o/μ-ratio that applies for this batch. The relationship between the τ_o/μ-ratio and the potential gravel migration, is discussed in Section 10.3.1.

In the right illustration of Figure 10.11, the value $100 \cdot (f_o - f_{disc})/f_o$ increases or remains constant with decreasing f_o. This type of result is often produced and seems to be one of the characteristics of gravel migration. This trend is reversed for the Bingham case, where $100 \cdot (f_o - f_b)/f_o$ is actually decreasing with decreasing rotational frequency f_o of the outer cylinder.

Figure 10.12 displays the viscometric values $\tau_o^{\diamondsuit} \wedge \mu^{\diamondsuit}$ of the remaining fat concrete sample within the $r = R_i + 2\,\text{cm}$ circle (marked with $\diamondsuit$). This calculation is done in the same manner as is made for Figure 10.10. The same findings are made here as is done for the last-mentioned figure. The yield value of the mortar $\triangle$ is lower than of the remaining fat concrete sample within the $r = R_i + 2\,\text{cm}$ circle $\diamondsuit$. This indicates that this fat concrete is not depleted of gravel particles. Also, the results designated as "$R_o = 14.5\,\text{cm}$" and marked with $\bigcirc$ and $\square$ are always measured with a lower yield value than measured for the fat concrete $\diamondsuit$ within the circle of $r = R_i + 2\,\text{cm}$.

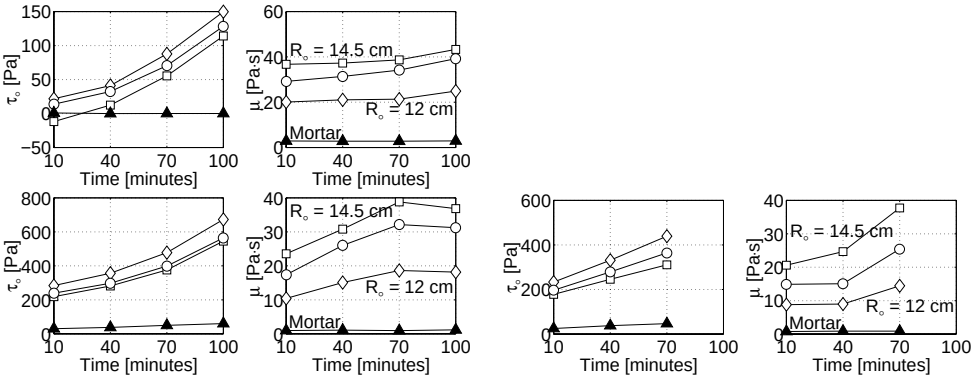


Figure 10.12: Results from the different ways in calculating the viscometric values τ_o and μ of the fresh concrete: $\tau_o^{\square} \wedge \mu^{\square}$; $\tau_o^{\bigcirc} \wedge \mu^{\bigcirc}$; $\tau_o^{\diamondsuit} \wedge \mu^{\diamondsuit}$. Mix design: **VHMW Na** at [0.6% sbwc; $w/c = 0.4$] (top), [0.3% sbwc; $w/c = 0.5$] (left) and [0.1% sbwc; $w/c = 0.6$] (right).

In relation to Figure 10.11, with the results at 10, 40 and 70 minutes after water addition, one can see how the fluctuating torque (i.e. the amplitude of the random oscillations in torque) are diminishing with time. This does not have to be due to the depletion of gravel particles within the circle, but rather due to the lower solid

particle interactions as a result of lower rotational frequency f_o. This is apparent when looking at how the fluctuation increase again between 77 and 88 seconds, when the rotational frequency f_o is increased from 0.09 rps upto 0.33 rps.

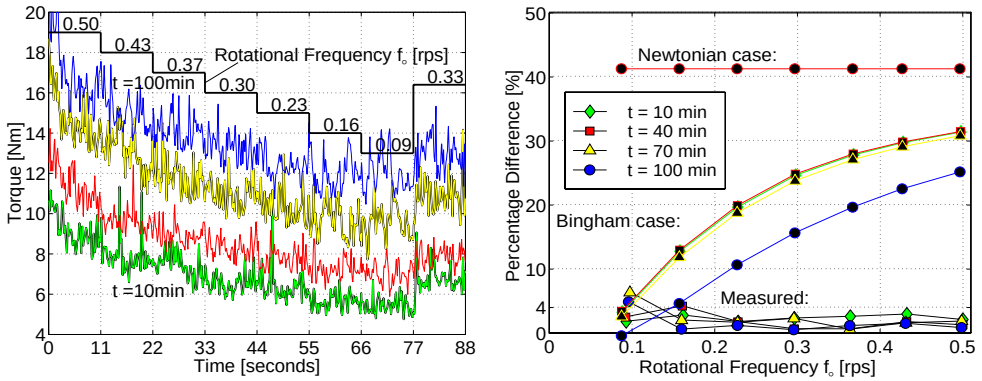


Figure 10.13: Test sample: Fresh concrete, using 0.3% sbwc of **VHMW Na** at $w/c = 0.5$. To the left is the measured torque $\hat{T}$ and to the right is the percentage difference between rotational frequency at $r = R_i + 2\,\mathrm{cm}$ and the rotational frequency of a solid f_o.

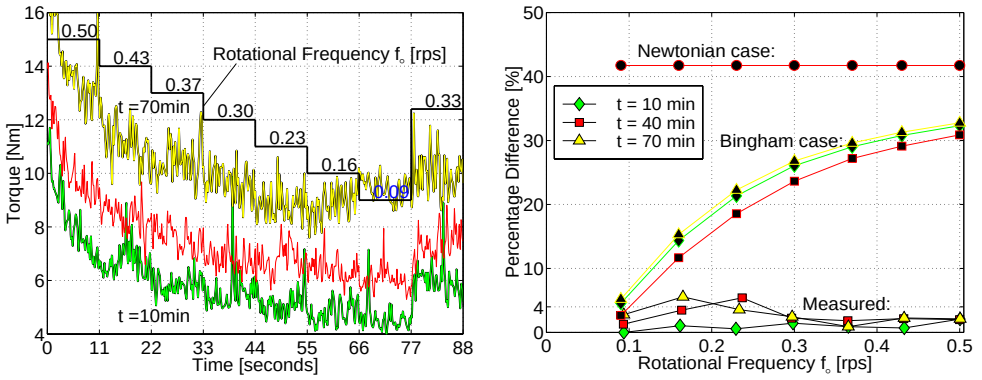


Figure 10.14: Test sample: Fresh concrete, using 0.1% sbwc of **VHMW Na** at $w/c = 0.6$. To the left is the measured torque $\hat{T}$ and to the right is the percentage difference between rotational frequency at $r = R_i + 2\,\mathrm{cm}$ and the rotational frequency of a solid f_o.

Figures 10.13 and 10.14 shows rheological results from fresh concrete of [0.3% sbwc; $w/c = 0.5$] and [0.1% sbwc; $w/c = 0.6$] using the **VHMW Na** polymer. The yield values produced for the $w/c = 0.5$ case are $\tau_o^{\square} = 219, 281, 376$ and $546\,\mathrm{Pa}$ and likewise, the plastic viscosities produced consist of $\mu^{\square} = 23.6, 30.8, 38.8$ and $36.8\,\mathrm{Pa}\cdot\mathrm{s}$. For the $w/c = 0.6$ case, the viscometric values produced[4] are $\tau_o^{\square} = [178, 246, 311]\,\mathrm{Pa}$ and $\mu^{\square} = [20.6, 24.7, 37.7]\,\mathrm{Pa}\cdot\mathrm{s}$. As shown with the right illustration of either figure, the difference $100\cdot(f_o - f_b)/f_o$ is below 4% for most cases, indicating a plug condition in the vicinity of the circle of $r = R_i + 2\,\mathrm{cm}$.

Figure 10.15 demonstrates an additional experiment that is done. It consists of using the same mix design as in Figure 10.14, except for that the 11−16 mm aggregates

[4]For the measurement at 70 minutes after water addition, the logged torque data $\hat{T}$ between 66 and 77 seconds, are ignored in the data processing when generating the viscometric values.

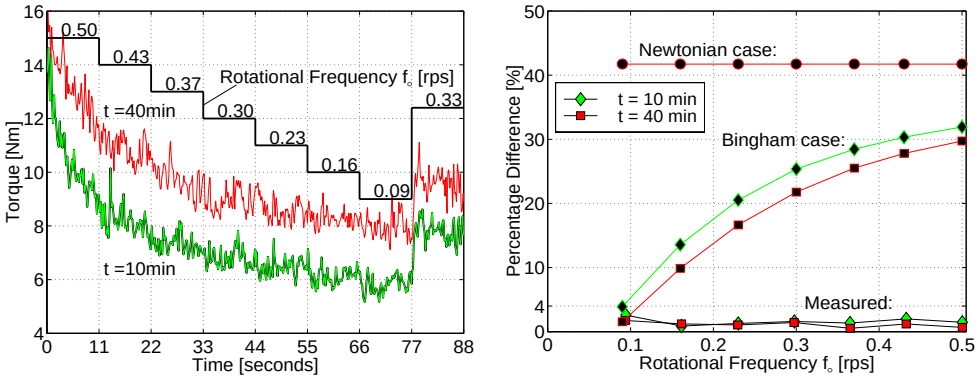


Figure 10.15: Same type of batch as in Figure 10.14, except for that the $11-16\,\mathrm{mm}$ aggregates are replaced with the $8-11\,\mathrm{mm}$ aggregates, giving now $D_{\max} = 11\,\mathrm{mm}$.

are replaced with the $8-11\,\mathrm{mm}$ aggregates, giving now $D_{\max} = 11\,\mathrm{mm}$. In doing this, the value of $D_{\mathrm{flow}}/D_{\max} = (R_{\mathrm{o}} - R_{\mathrm{i}})/D_{\max}$ is increased from 2.8 to 4. As is apparent with the right illustration, regardless of the increased motional freedom (for the larger solid particles), the gravel migration still occurs (see Section 3.4.2, about the "motional freedom" of the aggregates).

10.2.3 Summary and Conclusion (Part I)

The General Trend

Gravel migration $\mathbf{q}$ is not only governed by the gradient[5] in shear rate $-\nabla\dot{\gamma}$, but also by some internal and external properties, described with the migration coefficient ϑ. In the process of gravel migration, the remaining concrete, surrounding the inner cylinder R_{i}, becomes fatter and fatter. Its matrix content $1-\Phi$ increases, resulting in a decrease in its shear viscosity η. Hence, with Equations 3.12, 3.17 and 3.19 (Page 58), this will also result in the decrease of measured torque $\hat{T} = 2\,\pi\,R_{\mathrm{i}}^2\,h\,\eta(R_{\mathrm{i}})\,\dot{\gamma}(R_{\mathrm{i}})$, usually giving some kind of curvature in torque profile, as shown with Figure 10.18.

Measurements indicate that outside the circle of $r = R_{\mathrm{i}} + 2\,\mathrm{cm}$, the gravel concentration Φ reaches a saturation value $\Phi_{\max}$ within, say 10 - 15 seconds after start in rotation of the outer cylinder R_{o}. With the condition $\Phi = \Phi_{\max}$ outside this circle, a rigid body motion results. This is measured with the plastic disc: $f_{\mathrm{disc}} \approx f_{\mathrm{o}}$. After this time period, the gravel migration seems to continue only within this circle (i.e. between R_{i} and $R_{\mathrm{i}} + 2\,\mathrm{cm}$), for about additional 20 seconds or so. This is evident with the ongoing nonlinear decrease in torque $\hat{T}$ for this same time period.

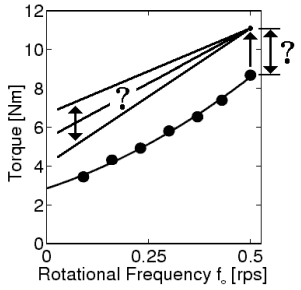
The measurements of the corresponding mortar mix, indicate only a thixotropic breakdown within the first 2.5 seconds of data logging (i.e. within the first 2.5 seconds of rotation of the outer cylinder). Hence, it could be suggested that thixotropic breakdown of the concrete should be finished well within the 2.5 seconds of rotation, or long before any data logging is made with the ConTec **BML** Viscometer 3 (c.f. the pre-rotation in Table 3.1).

After processing the raw torque data as described in Section 3.4 and thereafter plotting the resulting discrete torque points $\hat{T}$ as a function of rotational frequency f_{o},

[5] As should be apparent with Equation 3.23 (Page 59), the gradient of shear rate $-\nabla\dot{\gamma}$ is dependent on overall viscometric values τ_{o} and μ of the test sample and on the rotational frequency f_{o}.

the results like presented with the left illustration of Figure 10.18 are produced. The type of curves shown there are just a direct consequence of the curves demonstrated, for example, in the left illustration of Figure 10.6.

Rheological Results Shown in Chapter 6



From the above, it should be clear that the viscometric values of the concrete measurements of this thesis are somewhat incorrect due to gravel migration. As shown with the schematic figure to the right, the yield values τ_o are measured lower than the true ones, but the effect on the plastic viscosity μ is unclear. When gravel migration occurs, the best way to process the logged torque data, is to dismiss the first torque points $\hat{T}$, when gravel concentration $\Phi = \Phi(r, z, t)$ is changing most rapidly. This usually means only using the five torque points $\hat{T}$, generated when the rotational frequency is going from $f_o = 0.37\,\text{rps}$ down to $f_o = 0.09\,\text{rps}$ (see Figure 10.18). However, in taking this approach, one is only extracting the viscometric values related to the remaining fat concrete (that surrounds the inner cylinder), and not of the concrete as a whole, in a homogenous state. The set of viscometric values $\tau_o^{\diamondsuit} \wedge \mu^{\diamondsuit}$ represents the most correct values for this fat concrete. The corresponding G and H values are calculated by linear regression of the pre-mentioned five torque points $\hat{T}$ versus the rotational frequency of the disc f_{disc}. Thereafter the radius of $R_o = 12\,\text{cm}$ is used when applying Equations 3.26 and 3.27 ($R_i = 10\,\text{cm} \wedge h = 19.9\,\text{cm}$). This is because the needles that are submerged into the concrete sample are located at $r = R_i + 2\,\text{cm}$. This approach has been previously used in relation to Figures 10.10 and 10.12 (marked with $\diamondsuit$).

As is evident with the results presented in Figures 10.6, 10.7, 10.11, 10.13, 10.14 and 10.15, the rotational frequency of the disc f_{disc} is often very close to the rotational frequency of the outer cylinder f_o. This result gives the possibility of extracting an accurate viscometric values for the remaining fat concrete, of batches where the information about f_{disc} is not available: First, the G and H values are calculated by linear regression of the above-mentioned five torque points $\hat{T}$ versus the rotational frequency of the outer cylinder f_o ($\approx f_{\text{disc}}$). Thereafter, the radius of $R_o = 12\,\text{cm}$ is used when applying Equations 3.26 and 3.27 ($R_i = 10\,\text{cm} \wedge h = 19.9\,\text{cm}$). This results in the viscometric values $\tau_o^{\clubsuit} \wedge \mu^{\clubsuit}$ ($\approx \tau_o^{\diamondsuit} \wedge \mu^{\diamondsuit}$).

To investigate the validity of $\tau_o^{\clubsuit} \wedge \mu^{\clubsuit}$ and $\tau_o^{\bigcirc} \wedge \mu^{\bigcirc}$ (see Section 3.4.2), relative to $\tau_o^{\diamondsuit} \wedge \mu^{\diamondsuit}$, number of four calculations are made, listed below. The result of these calculations are presented in Figure 10.16.

- **Calculation 1:** Plot of $[100 \cdot (\tau_o^{\diamondsuit} - \tau_o^{\bigcirc})/\tau_o^{\diamondsuit}]$ versus $[\tau_o^{\bigcirc}/\mu^{\bigcirc}]$
- **Calculation 2:** Plot of $[100 \cdot (\tau_o^{\diamondsuit} - \tau_o^{\clubsuit})/\tau_o^{\diamondsuit}]$ versus $[\tau_o^{\clubsuit}/\mu^{\clubsuit}]$
- **Calculation 3:** Plot of $[100 \cdot (\mu^{\diamondsuit} - \mu^{\bigcirc})/\mu^{\diamondsuit}]$ versus $[\tau_o^{\bigcirc}/\mu^{\bigcirc}]$
- **Calculation 4:** Plot of $[100 \cdot (\mu^{\diamondsuit} - \mu^{\clubsuit})/\mu^{\diamondsuit}]$ versus $[\tau_o^{\clubsuit}/\mu^{\clubsuit}]$

Figure 10.16 shows that there is a small difference between the viscometric values $\tau_o^{\diamondsuit} \wedge \mu^{\diamondsuit}$ and $\tau_o^{\clubsuit} \wedge \mu^{\clubsuit}$. This is to be expected, since $f_{\text{disc}} \approx f_o$ applies for most cases. As such, the values $\tau_o^{\clubsuit} \wedge \mu^{\clubsuit}$ represent a better result, relative to the values $\tau_o^{\bigcirc} \wedge \mu^{\bigcirc}$ when extracting the viscometric values of the remaining fat concrete surrounding

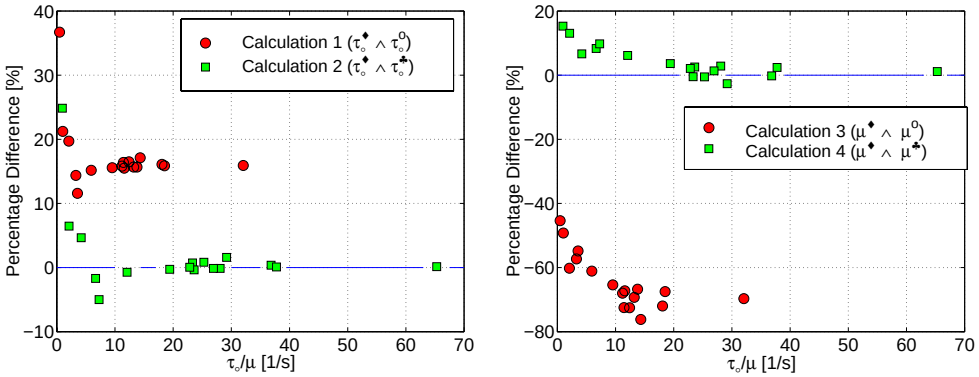


Figure 10.16: Percentage difference between viscometric values, processed in a different manner.

the inner cylinder. Figure 10.17 demonstrates a plot of the slump value (the actual response of the fresh concrete as a whole) versus the two viscometric values $\tau_o^{\clubsuit}$ and $\mu^{\clubsuit}$ of the fat concrete that surrounds the inner cylinder. This figure shows that although the two values $\tau_o^{\clubsuit}$ and $\mu^{\clubsuit}$ do not represent the viscometric response of the fresh concrete as a whole, there is a very good relationship between the yield value $\tau_o^{\clubsuit}$ and slump. There is however a weaker relationship between slump and the plastic viscosity $\mu^{\clubsuit}$. The same type of plot is also available for $w/c = 0.5$ and $w/c = 0.6$ in Chapter 6. For further discussion about the overall fine quality of the rheological results made by the ConTec **BML** Viscometer 3, see Section 6.3.

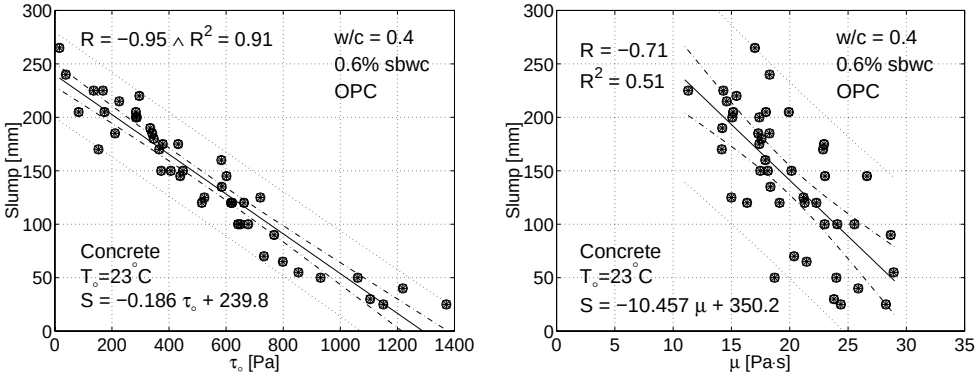


Figure 10.17: Comparison of measured slump value versus the yield value $\tau_o^{\clubsuit}$ (to the left) and plastic viscosity $\mu^{\clubsuit}$ (to the right). These results are extracted from Figure 6.6 (Page 108).

Herschel-Bulkley Model?

When making a detailed analysis on the fresh concrete, the effect of gravel migration will inflect and damage the viscometric results. Such typical analysis would be: **1)** When modulating a new type of flow model. A non-linear torque $\hat{T}$ generated by gravel migration, could incorrectly lead to the suggestion that a new material model applies, for example of the Herschel-Bulkley model $\eta = \mu\,\dot{\gamma}^{n-1} + \tau_o/\dot{\gamma}$. **2)** When trying to measure a thixotropic behavior in the fresh concrete. Doing a *stepwise increasing*

shear rate sequence followed by a *stepwise decreasing shear rate sequence*, produces a hysteresis loop, which could be understood as thixotropic breakdown, when in fact it is also a result of continuous gravel migration. If occurring simultaneously, separating the two is a challenging task. An interesting discussion about the two phenomena when occurring simultaneously, is made by Wesche et al. [149]. **3)** Often, it is interesting to create a shear viscosity function of a suspension $\eta = \eta(\eta_{\mathrm{m}})$, dependent on viscometric values of its matrix phase η_{m} (i.e. creating a particle-matrix model). However, such model can be somewhat damaged, when gravel migration is occurring.

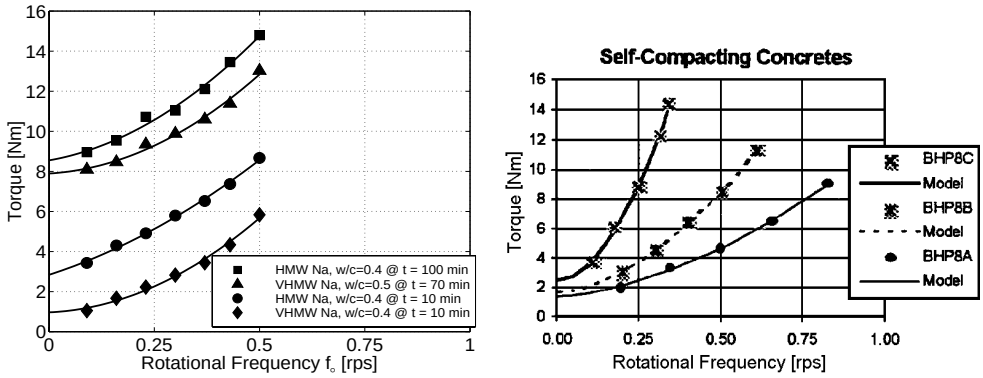


Figure 10.18: Regression analysis assuming the Herschel-Bulkley model applies. To the left are torque points used from the figures in Section 10.2. To the right is data presented by Ferraris and de Larrard [31], produced with the BTRHEOM viscometer. For both illustrations, the *stepwise decreasing shear rate sequence* is applied.

As mentioned in the introduction of this chapter, gravel migration can also occur in other types of viscometers, for example the BTRHEOM (see Section 10.3.4). In fact, for this last-mentioned viscometer, a curve in torque $\hat{T}$ has been produced as shown with the right illustration of Figure 10.18. For such instances, it is concluded that fresh concrete corresponds to the Herschel-Bulkley model [26, 31].

10.3 Suggestions to Geometrical Changes

In mathematical terms, gravel migration can be stated with $\mathbf{q} = -\vartheta\,\nabla\dot{\gamma}$, where the migration coefficient $\vartheta = \vartheta(\Phi, \eta_{\mathrm{m}}, D_{\mathrm{flow}}/D_{\mathrm{max}}, \dots)$ describes how much amount of gravel should migrate $\mathbf{q}$ for the given change in shear rate $\nabla\dot{\gamma}$. Using this approach, it is apparent that the direction and the potential magnitude of gravel migration is controlled by the gradient of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}$, plotted as arrows in Figure 10.1. The direction of gravel migration as predicted in this figure, is experimentally observed as is shown in Figures 10.2 and 10.3. With this in mind, one can explore the possibility for a more suitable viscometric geometry to the ConTec **BML** Viscometer 3, in minimizing the probability of gravel migration; i.e. the following analysis will be based on the gradient of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}$, where the geometry that has the lowest difference in shear rate (i.e. has the has smallest vectors), is considered to be the best solution. Like before, the numerical calculation conducted in this chapter, consist of solving the problem described with Equations 7.58 (or equally 7.6) and 7.59 on Page 167.

10.3.1 Modifying the Current Configuration

Fortunately, it is possible to reduce the gradient in shear rate $-\nabla\dot{\gamma}$ inside the coaxial cylinders viscometers and hence lessen the possibility for gravel migration. However, the magnitude in this reduction depends very much on the ratio of yield value to plastic viscosity τ_o/μ. Applying Equations 3.29 and 3.32 together (see Pages 60 and 66) produces results like shown in Figure 10.19: Using $\tau_o/\mu = 10\,\mathrm{s}^{-1} \wedge \omega_o = 3\,\mathrm{rad/s}$ ($f_o = 0.48\,\mathrm{rps}$) and then changing the geometry from $(R_i, R_o) = (10, 14.5)\,\mathrm{cm}$ to $(R_i, R_o) = (14, 22)\,\mathrm{cm}$ [(b) $\rightarrow$ (d)], leads to a slight reduction in shear rate gradient $-\nabla\dot{\gamma} = -d\dot{\gamma}/dr\,\mathbf{i}_r$. Reducing the angular velocity down to $\omega_o = 0.5\,\mathrm{rad/s}$ ($f_o = 0.08\,\mathrm{rps}$) does not generate as much improvements as one would like, due to the concomitant plug formation [(b) $\rightarrow$ (a) $\wedge$ (d) $\rightarrow$ (c)]. However, if the test material consist of $\tau_o/\mu = 1\,\mathrm{s}^{-1}$ [(e),(f),(g) $\wedge$ (h)], plug does not occur and hence reducing the angular velocity ω_o is very beneficial [(f) $\rightarrow$ (e) $\wedge$ (h) $\rightarrow$ (g)].

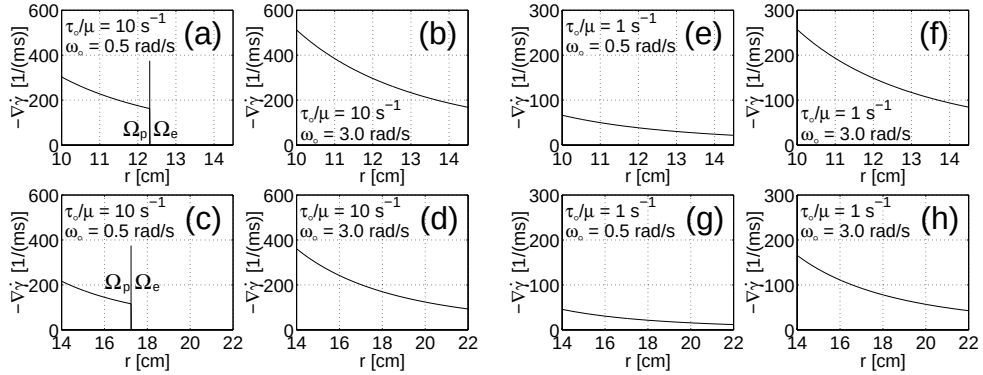


Figure 10.19: Gradient of shear rate $-\nabla\dot{\gamma}$ inside a coaxial cylinder viscometer using different geometry $R_i \wedge R_o$ and using different ratio of yield value and plastic viscosity τ_o/μ. The case of illustration (b) is also present in Figure 10.1.

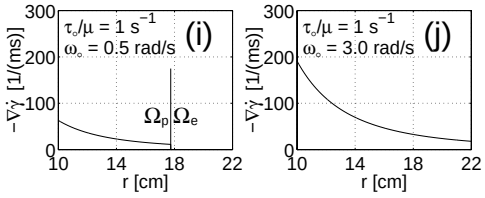


Figure 10.20: A continuation of Figure 10.19.

It is interesting to note how heavily the gradient $-\nabla\dot{\gamma}$ is dependent on the ratio τ_o/μ. Decreasing this ratio from $\tau_o/\mu = 10\,\mathrm{s}^{-1}$ down to $\tau_o/\mu = 1\,\mathrm{s}^{-1}$, reduces the gradient by about half if plug is not present [(b) $\rightarrow$ (f) $\wedge$ (d) $\rightarrow$ (h)]. If plug is occurring for the $\tau_o/\mu = 10\,\mathrm{s}^{-1}$-case, then this reduction becomes even larger [(a) $\rightarrow$ (e) $\wedge$ (c) $\rightarrow$ (g)]. With this information in mind, it is possible to solve the problem of gravel migration when the ratio τ_o/μ is sufficiently low, say below $2\,\mathrm{s}^{-1}$. Such condition applies frequently for self-compacting concrete. The potential for gravel migration could be further reduced by also reducing the maximum value of the rotational frequency, from $f_{max} = 0.5\,\mathrm{rps}$ down to $f_{max} = 0.25\,\mathrm{rps}$. However in making this step, care must be

taken to ensure that the smallest time possible, passes between the end of (re)mixing and the start of a measurement. If not, thixotropic effects could damage the result. For the $\tau_o/\mu = 1\,s^{-1}$-case, there is only minor reduction in the shear rate gradient $-\nabla\dot{\gamma} = -d\dot{\gamma}/dr\,\mathbf{i}_r$, when changing the geometry from $(R_i, R_o) = (10, 14.5)\,cm$ to $(R_i, R_o) = (14, 22)\,cm$ [(e) $\rightarrow$ (g) $\wedge$ (f) $\rightarrow$ (h)]. As such, there does not seem to be much benefit in using the larger geometry.

As stated in Section 10.1, gravel migration is partly related to the difference in collisions rate $-\nabla\dot{n}$, which is again related to the difference in shear rate $-\nabla\dot{\gamma}$. Consequently, the gravel particles are "pushed" by collisions, towards the outer cylinder and also into the serrated region of inner cylinder, where the shear rate $\dot{\gamma}$ has the smallest value. With the relative small gap system of $(R_i, R_o) = (10, 14.5)\,cm$, which results in the condition $D_{flow}/D_{max} \approx 3$, the gravel particles could collide more strongly with each other due to their lack of motional freedom in avoiding such a direct and strong mechanical interactions (see Section 3.4.2). This could result in a stronger "pushing mechanism", than first anticipated. If this is the case, increasing the gap between the inner and outer cylinder $D_{flow} = R_o - R_i$ would be much more beneficial than indicated by the steps of (e) $\rightarrow$ (g) and (f) $\rightarrow$ (h), shown in Figure 10.19.

10.3.2 Exploring Cone and Parallel Plates Geometries

It is relatively easy to apply different viscometric geometries to the ConTec **BML** Viscometer 3. An example of such modification is shown in Figures 10.21 and 10.31 (see also Figure 3.4, Page 55). With this option, it is only natural to explore other geometrical possibilities in minimizing the potential for gravel migration $\mathbf{q} = -\vartheta\nabla\dot{\gamma}$. This is the subject of Sections 10.3.4, 10.3.5, 10.3.6 and 10.3.7, which now follows. Unless otherwise stated, for all figures represented in these sections, the following parameters apply: $\mu = 20\,Pa \cdot s$, $\tau_o = 200\,Pa$, $\delta = 4 \cdot 10^{-3}\,s^{-1}$ and $\omega_o = \omega_t = 3\,rad/s$. The dashed dotted line for these figures, always demonstrates the boundary between a solid state Ω_e and a viscoplastic state Ω_p, plotted with a single isoline of $\tau_c = 200\,Pa$. The above values are also used for the result shown in Figure 10.1. For all figures that demonstrate an isoplot of shear rate $\dot{\gamma}_c$, the following values apply for the isolines: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12.5, 15, 17.5, 20, 22.5, 25, 27.5, 30]\,s^{-1}$. The arrows also shown for such cases, are the vectors of the shear rate gradient $-\nabla\dot{\gamma}$. The reference vector, displayed inside a corresponding geometry of each such figure, always consists of $\|\nabla\dot{\gamma}\| = 10^3\,m^{-1}s^{-1}$. This is done to ease the comparison between figures.

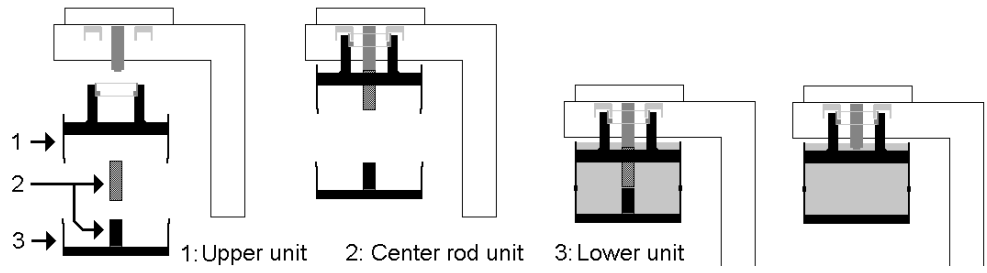


Figure 10.21: The assembly of BTRHEOM-geometry (a kind of parallel plates geometry) to the ConTec **BML** Viscometer 3. An enhancement of this geometry would simply consist of removal of the center rod unit as is shown with the most right illustration.

10.3.3 Velocity Profile

By using the general velocity field $\mathbf{v} = v_\mathrm{r}(r,\theta,z,t)\,\mathbf{i}_\mathrm{r} + v_\theta(r,\theta,z,t)\,\mathbf{i}_\theta + v_\mathrm{z}(r,\theta,z,t)\,\mathbf{i}_\mathrm{z}$, a very complex system- and a very large number of algebraic equations are produced, when converting Equation 2.17 (Page 16) to its algebraic counterpart. This means that the calculation time becomes very large, allowing only handful of numerical analysis to be made for the given computer hardware available. But fortunately, some reasonable assumptions about the flow can be made, resulting in Equation 10.1. This reduces the number of algebraic equations down to what is shown with Equations 7.26 and 7.27 (Page 161) and makes each calculation a great deal faster. Equation 10.1 is extracted from the following items. These items are assumed to be valid for all the viscometers presented in Sections 10.3.4, 10.3.5, 10.3.6 and 10.3.7.

1. With low Reynolds number (i.e. with low speed and high shear viscosity η) the flow is stable[6] and it is possible to assume a flow symmetry around the z-axis: $\mathbf{v} = v_\mathrm{r}(r,\theta,z,t)\,\mathbf{i}_\mathrm{r} + v_\theta(r,\theta,z,t)\,\mathbf{i}_\theta$.

2. Due to the circular geometry of the parallel plates and of the cylinder geometries that are involved (see for example Figures 10.23 and 10.30), it is reasonable to assume pure circular flow with θ-independence:

$$\mathbf{v} = v_\theta(r,z,t)\,\mathbf{i}_\theta \tag{10.1}$$

With the above velocity field, all the results of Chapter 7 can be applied.

10.3.4 BTRHEOM-Geometry

Solution Geometry and Boundary Condition

The left illustration of Figure 10.22 demonstrates the solution geometry and boundary conditions used when simulating the BTRHEOM viscometer. The solution array is 121×101 in size and the spacing of grid points in r- and z-direction are uniform and equal $\Delta r = \Delta z = 1.0\,\mathrm{mm}$. As mentioned in Chapter 7, this greatly simplifies the programming of the solution and save storage space. Usually, this also results in greater accuracy [2]. In the solution algorithm, the viscometer is rotated by 90° clockwise to simplify array addressing. This is apparent with the right illustration of Figure 10.22 and is perhaps more apparent with Figure 10.23, which demonstrates a three dimensional vector plot of velocity $\mathbf{v}$ inside the BTRHEOM viscometer (no-slippage-case). As shown, it consist of a rotating top part (that measures applied torque from the test material) and a stationary bottom part.

Referring to the left illustration of Figure 10.22, at the top- and bottom plates (i.e. at $(\mathrm{i},\mathrm{j}) = (21:121,101) \wedge (\mathrm{i},\mathrm{j}) = (21:121,1)$), the Dirichlet boundary conditions $v_\theta = r\,\omega_\mathrm{t}$ and $v_\theta = 0$ applies, respectively. These conditions are valid for both the no-slippage-case and the full-slippage-case.

The outer cylinder R_o and the inner cylinder R_i (located at $(\mathrm{i},\mathrm{j}) = (121,1:101) \wedge (\mathrm{i},\mathrm{j}) = (21,1:101)$), will be designated as the "wall boundary". At this wall boundary, one of two different types of boundary condition applies. First is the Dirichlet boundary condition, consisting of $v_\theta = R_\mathrm{o}\,\omega_\mathrm{t}$ for the rotating part of the "right wall" and of $v_\theta = R_\mathrm{i}\,\omega_\mathrm{t}$ for the rotating part of the "left wall" (see the left illustration of Figure 10.22). The rest of the wall boundary has the boundary condition $v_\theta = 0$.

[6] See Footnote 8 on Page 56.

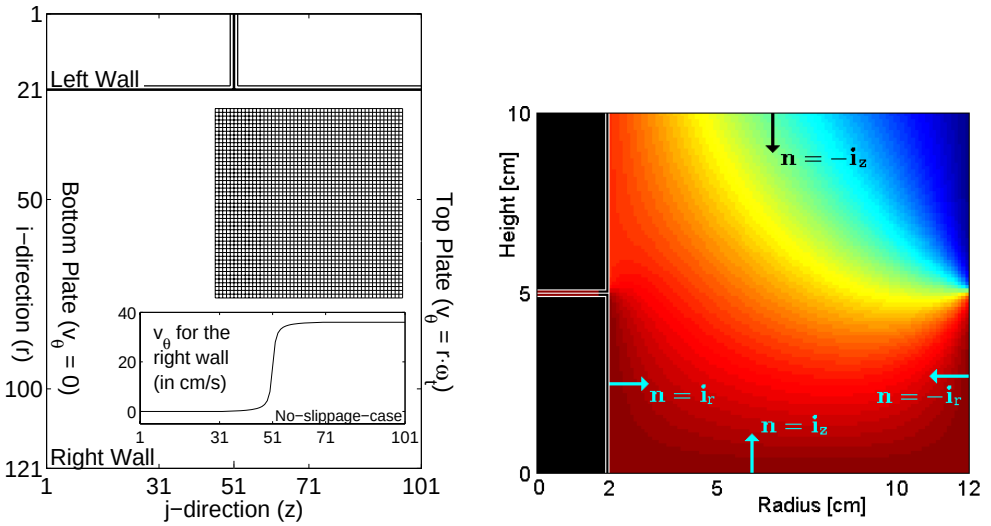


Figure 10.22: To the left: Solution geometry for the BTRHEOM viscometer. The upper right corner demonstrates the grid density used. To the right: A relative velocity profile, with the unit vectors of the wall boundaries and of the bottom and top plates. $[(\tau_o, \mu) = (200, 20\text{s})\text{Pa}; \omega_t = 3\text{rad/s}; \delta = 4 \cdot 10^{-3}\text{s}^{-1}; (\Delta r, \Delta z) = (1, 1)\text{mm}; \Delta t = 4 \cdot 10^{-9}\text{s}]$.

These conditions apply when the no-slippage-case (ns) is valid. For the full-slippage-case (fs), the boundary condition is of the Robin type, described with Equations 7.56 and 7.57 (Page 167). When used, this condition applies equally to all parts of the wall boundary.

For the no-slippage-case (ns), then at the right wall (as well for the left wall), the rotating part $v_\theta = R_o\,\omega_t$ and the stationary part ($v_\theta = 0$) are connected with a *arctan* function. As shown with the left illustration of Figure 10.22, this connection occurs at the points $(\text{i}, \text{j}) = (121, 31)$ and $(\text{i}, \text{j}) = (121, 71)$ (similar applies for the left wall). The *arctan* function is used, to have a smooth transition between the rotating part and the non-rotating part of the viscometer.

The total torque applied <u>from</u> the test material, <u>on</u> the rotating part (i.e. on the measuring unit) of the viscometer, is $\hat{\mathbf{T}} = \hat{\mathbf{T}}_{\text{lw}} + \hat{\mathbf{T}}_{\text{tp}} + \hat{\mathbf{T}}_{\text{rw}}$. The term $\hat{\mathbf{T}}_{\text{tp}}$ is the torque applied <u>from</u> the test material, <u>on</u> the top plate. With the condition of $(\tau_o, \mu) = (200, 20\text{s})\text{Pa}$ and $\omega_t = 3\,\text{rad/s}$, this term is calculated with the following:

$$\hat{\mathbf{T}}_{\text{tp}} = \int_{R_i}^{R_o} \int_0^{2\pi} r\,\mathbf{i}_r \times (-\mathbf{i}_z \cdot \mathbf{T}|_{z=h}\, r\, d\theta\, dr) = \begin{cases} -0.48\,\text{Nm}\,\mathbf{i}_z\ (\text{ns}) \\ -0.92\,\text{Nm}\,\mathbf{i}_z\ (\text{fs}) \end{cases} \tag{10.2}$$

The normal vector used in the above equation is shown in the right illustration of Figure 10.22. The designation <u>ns</u> means no-slippage-case and <u>fs</u> means full-slippage-case. Since $\mathbf{i}_r \times (-\mathbf{i}_z \cdot \mathbf{T}) \leq 0$ in the above, the resulting torque value becomes negative[7]. This is to be expected since the test material is always trying to slow down the rotating part of the viscometer. The same consideration applies for the two following equations.

[7] A positive torque value would be gained if the torque applied <u>from</u> the rotating part (i.e. from the measuring unit) of the viscometer, <u>on</u> the test material would be calculated (since $\mathbf{i}_r \times (\mathbf{i}_z \cdot \mathbf{T}) \geq 0$). Nevertheless, the same absolute torque value is gained in either case.

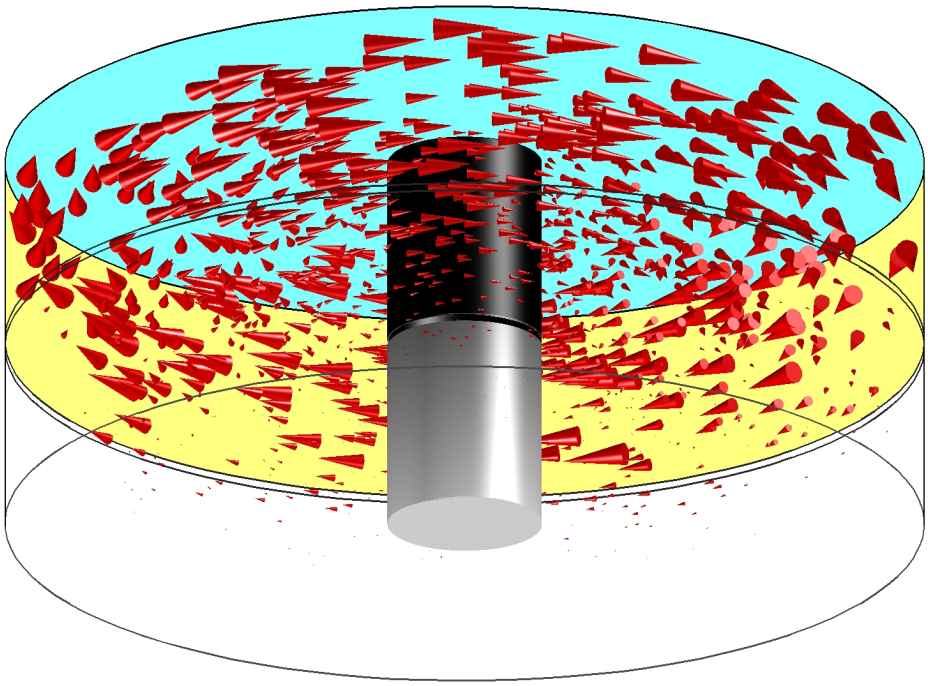


Figure 10.23: Three dimensional vector plot of velocity $\mathbf{v}$, inside the BTRHEOM viscometer when the no-slippage-case applies. $[(\tau_o, \mu) = (200, 20\text{s})\text{Pa}$; $\omega_t = 3\text{rad/s}$; $\delta = 4 \cdot 10^{-3}\text{s}^{-1}$; $(\Delta r, \Delta z) = (1, 1)\text{mm}$; $\Delta t = 4 \cdot 10^{-9}\text{s}]$.

The term $\hat{\mathbf{T}}_{\text{lw}}$ represents is the torque applied from the test material, on the (upper and rotating) left wall. This term is calculated by the following:

$$\hat{\mathbf{T}}_{\text{lw}} = \int_{h/2}^{h} \int_{0}^{2\pi} R_i \, \mathbf{i}_r \times (\mathbf{i}_r \cdot \mathbf{T}|_{r=R_i} \, R_i \, d\theta \, dz) = \begin{cases} -0.03\,\text{Nm}\,\mathbf{i}_z \text{ (ns)} \\ 0\,\text{Nm}\,\mathbf{i}_z \text{ (fs)} \end{cases} \tag{10.3}$$

Finally, the term $\hat{\mathbf{T}}_{\text{rw}}$ is the torque applied from the test material, on the (upper and rotating) right wall. It is calculated by Equation 10.4.

$$\hat{\mathbf{T}}_{\text{rw}} = \int_{h/2}^{h} \int_{0}^{2\pi} R_o \, \mathbf{i}_r \times (-\mathbf{i}_r \cdot \mathbf{T}|_{r=R_o} \, R_o \, d\theta \, dz) = \begin{cases} -1.05\,\text{Nm}\,\mathbf{i}_z \text{ (ns)} \\ 0\,\text{Nm}\,\mathbf{i}_z \text{ (fs)} \end{cases} \tag{10.4}$$

For the no-slippage-case of $(\tau_o, \mu) = (200, 20\text{s})\text{Pa}$ and $\omega_t = 3\,\text{rad/s}$, the total torque applied on the rotating part of the viscometer is equal to $\hat{\mathbf{T}} = \hat{\mathbf{T}}_{\text{lw}} + \hat{\mathbf{T}}_{\text{tp}} + \hat{\mathbf{T}}_{\text{rw}} = -1.56\,\text{Nm}\,\mathbf{i}_z$. For the full-slippage-case the total torque is $-0.92\,\text{Nm}\,\mathbf{i}_z$ (calculated either by the software or by Equation B.51). It is this rotating part of the BTRHEOM viscometer that measures the torque. As such, this device should measure a 70% increase in torque when going from the full-slippage-case to the no-slippage-case. This result is in contradiction to the finite element analysis made by Hu et al. [51]: With the same conditions of $(\tau_o, \mu) = (200, 20\text{s})\text{Pa}$ and $\omega_t = 3\,\text{rad/s}$ as in the above, it is demonstrated that a very similar torque values should be measured for both the full-slippage-case and the no-slippage-case (i.e. close to $0.92\,\text{Nm}$).

As a rough quality check of the numerical calculation done in this section, the total torque applied from the test material on the viscometer is calculated, both on

the rotating part (the measuring unit) and on the stationary part (the non-measuring unit). Since the test material is not an engine, it cannot perform a work by it self and hence the total torque from the test material on the viscometer should be zero. For the no-slippage-case, this value is equal to $-1.36 \cdot 10^{-11}\,\mathrm{Nm}\,\mathbf{i}_z$, while for the full-slippage-case, it consist of $-6.15 \cdot 10^{-15}\,\mathrm{Nm}\,\mathbf{i}_z$. Relative to the applied torque on the rotating part of the viscometer, this means $1.36 \cdot 10^{-11}\,\mathrm{Nm}/1.56\,\mathrm{Nm} \approx 10^{-11}$ and $6.15 \cdot 10^{-15}\,\mathrm{Nm}/0.92\,\mathrm{Nm} \approx 10^{-14}$, respectively.

Figure 10.24 demonstrates computed shear rate $\dot{\gamma}_c$ for the no-slippage-case (to the left) and for the full-slippage-case (to the right). This computation is made from the same numerical result shown in Figures 10.22 and 10.23. When the no-slippage-case applies for this device, there is a very large potential for gravel migration. In fact, the length of the vectors $-\nabla\dot{\gamma}_c$ are the largest calculated in the overall analysis of Chapter 10. This means that formation of a shearing zone filled with fat concrete could occur in the zone of large vectors, while a measurement is being conducted. This migration would appear in the most critical area of the viscometer: As is shown with Equation 10.4 (ns), the largest torque is generated in the vicinity of this zone, resulting in a large influence of gravel migration on logged torque $\hat{T}$. If occurring, the result presented in Figure 10.18 would most likely be produced.

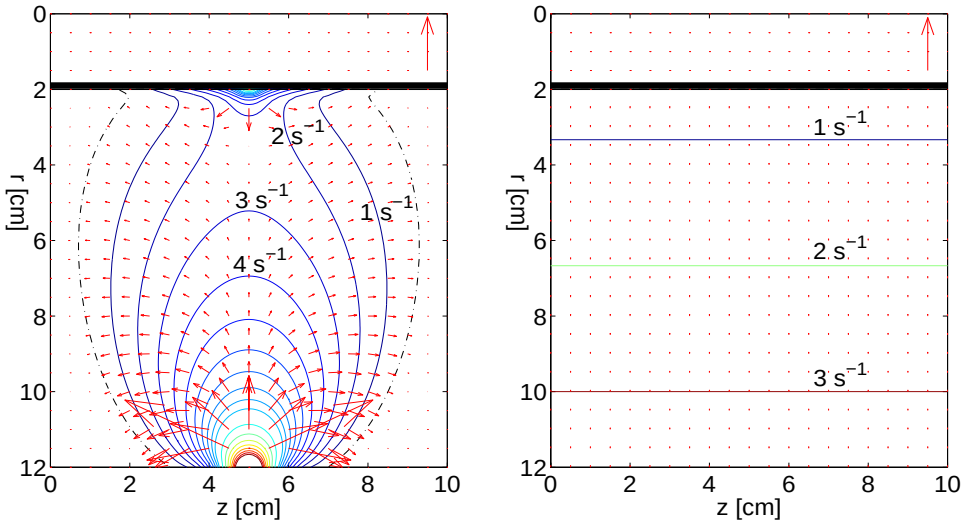


Figure 10.24: Gradient field of shear rate: $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_c$. The length of the vector, shown inside the inner cylinder, is $\|\nabla\dot{\gamma}\| = 10^3\,\mathrm{m}^{-1}\mathrm{s}^{-1}$. The isoplot is of shear rate $\dot{\gamma}_c$ and the dashed dotted line shows the boundary between a solid state and a viscoplastic state. $[(\tau_o, \mu) = (200, 20\mathrm{s})\mathrm{Pa}$; $\omega_t = 3\mathrm{rad/s}$; $\delta = 4 \cdot 10^{-3}\mathrm{s}^{-1}$; $(\Delta r, \Delta z) = (1, 1)\mathrm{mm}$; $\Delta t = 4 \cdot 10^{-9}\mathrm{s}]$.

For the full-slippage-case (the right illustration of Figure 10.24), the ideal and the most wanted shear rate condition is achieved. The gradient of shear rate is virtual zero $-\nabla\dot{\gamma}_c \approx 0$, and hence virtual no potential for gravel migration is present. The computed shear rate profile $\dot{\gamma}_c$ shown there, is identical to the analytical shear rate $\dot{\gamma}$, as calculated with Equation B.49 (Page 393).

The full-slippage-case inside the BTRHEOM-geometry, is allegedly [51] implemented by using a smooth surface at the wall boundary. With a very adhesive material like that of the fresh concrete, such a boundary condition is not necessarily achieved. It is evident from the above analysis that there is a large difference between

the full-slippage-case and the no-slippage-case, relative to the potential of gravel migration and also relative to the applied torque $\hat{\mathbf{T}} = \hat{\mathbf{T}}_{\mathrm{lw}} + \hat{\mathbf{T}}_{\mathrm{tp}} + \hat{\mathbf{T}}_{\mathrm{rw}}$. Therefore, it is not considered a good idea to make the use of smooth surface in the analysis that now follow in the hope of generating the full-slippage boundary condition (for the corresponding experiment with concrete). Instead, geometry with rough or serrated surfaces is introduced at all solid boundaries, so that it is know with good confidence, that the no-slippage-case always applies.

10.3.5 Cone-Geometry

The left illustration of Figure 10.25 demonstrates the solution geometry and boundary conditions used when simulating the cone-type viscometer. For this particular figure, the solution array is 110×92 in size. As always, in the software algorithm, the viscometer is rotated by 90° clockwise to simplify array addressing. This is apparent when comparing the left part of Figure 10.25 with Figure 10.26.

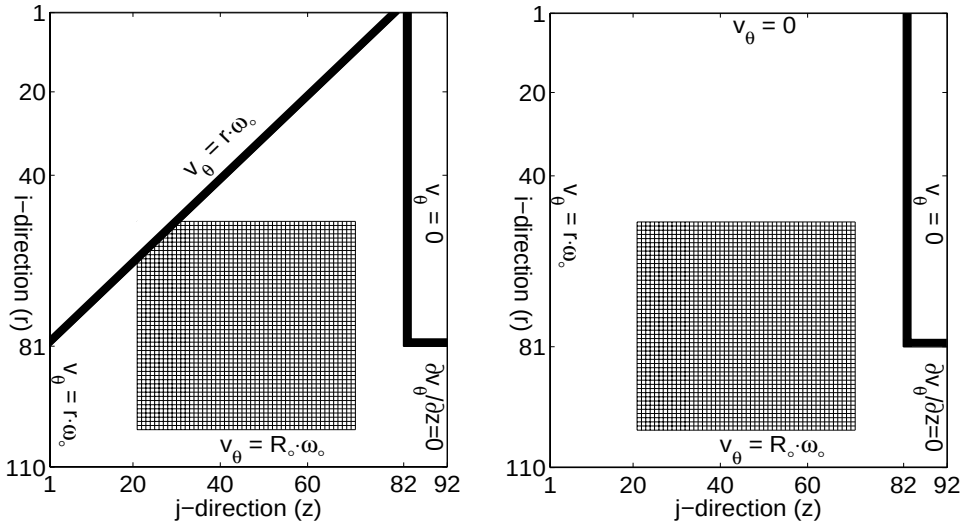


Figure 10.25: Solution geometry and boundary conditions for the cone viscometer (to the left) and the parallel plates viscometer (to the right). The lower part of both illustrations demonstrates the grid density used.

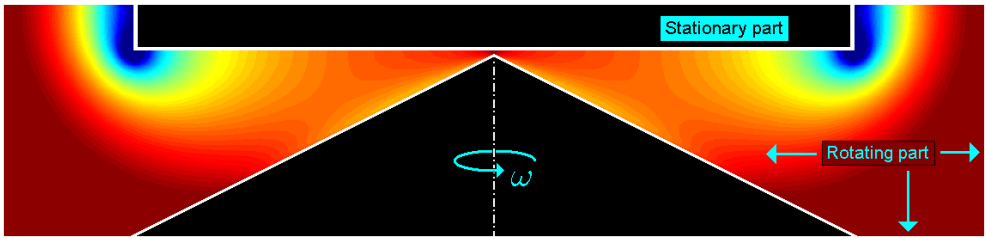


Figure 10.26: Computed shear rate profile $\dot{\gamma}_{\mathrm{c}}$ for the cone viscometer. This result is taken from the right illustration of Figure 10.27.

At the open boundary $(\mathrm{i}, \mathrm{j}) = (82 : 109, 92)$, the Neumann boundary condition of

$\partial v_\theta/\partial z = 0$ is used: The combination of high shear viscosity of the fresh concrete[8] and low rotational frequency of the viscometer $f_{\max} \approx 0.5\,\mathrm{rps}$, it is reasonable to assume that little or no inclination of the open boundary will be observed while making a rheological measurement. Therefore, the normal vector of this open boundary (i.e. at (i,j)=(82:109,92)) is set to $\mathbf{n} \cong \mathbf{i}_z$. Since practically no shearing is applied from the atmospheric air onto the concrete at this boundary, then according to Cauchy's stress principle [72], applying the vector $\mathbf{n} = \mathbf{i}_z$ on Equation 7.4 (Page 156), must result in zero: $\mathbf{i}_z \cdot \mathbf{T} = 0$. This means that the condition $\partial v_\theta/\partial z = 0$ becomes valid at the open boundary.

In the following discussion, then with the phrase "*the side of the top plate*", it is meant the boundary of $(\mathrm{i},\mathrm{j}) = (81, 82:92)$. Likewise with the phrase "*the bottom of the top plate*", it is meant the boundary of $(\mathrm{i},\mathrm{j}) = (1:81, 82)$. With "*the corner of the top plate*" it is meant the point of $(\mathrm{i},\mathrm{j}) = (81, 82)$ (see Figure 10.25).

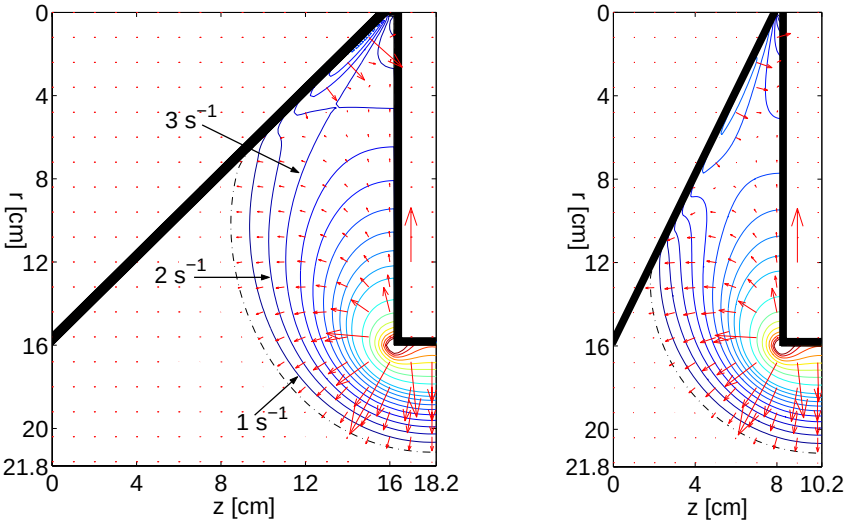


Figure 10.27: Gradient field of shear rate: $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_\mathrm{c}$ when the cone is 16 cm (to the left) and 8 cm (to the right) in height. The length of the vector, shown inside the top plate, is $\|\nabla\dot{\gamma}\| = 10^3\,\mathrm{m}^{-1}\mathrm{s}^{-1}$. The isoplot is of shear rate $\dot{\gamma}_\mathrm{c}$ and the dashed dotted line shows the boundary between a solid state and a viscoplastic state. $[(\tau_\mathrm{o}, \mu) = (200, 20\mathrm{s})\mathrm{Pa};\ \omega_\mathrm{o} = 3\mathrm{rad/s};\ \delta = 4 \cdot 10^{-3}\mathrm{s}^{-1};\ (\Delta r, \Delta z) = (2, 2 \wedge 1)\mathrm{mm};\ \Delta t = 10^{-6}\mathrm{s}$ (left) $\wedge\ 0.5 \cdot 10^{-6}\mathrm{s}$ (right)].

Figure 10.27, shows the gradient field of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_\mathrm{c}$ when the cone is $h_\mathrm{cone} = 16\,\mathrm{cm}$ (to the left) and $h_\mathrm{cone} = 8\,\mathrm{cm}$ (to the right) in height. The height of the latter cone is reduced by half, using $\Delta z = 1\,\mathrm{mm}$, instead of $\Delta z = 2\,\mathrm{mm}$. In both simulations the spacing of grid points in r-direction is $\Delta r = 2\,\mathrm{mm}$. The isoplot is of shear rate $\dot{\gamma}_\mathrm{c}$ and the smallest isoline is near to the outer cylinder and increases towards the corner of the top plate.

What is immediately apparent for both illustrations in Figure 10.27 is how small the gradients $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_\mathrm{c}$ are, below the bottom of the top plate. This means that there is a little potential for gravel migration in this zone. However, in both cases of $h_\mathrm{cone} = 16\,\mathrm{cm}$ or $h_\mathrm{cone} = 8\,\mathrm{cm}$, there is a large potential for gravel migration in the zone around the corner and near the side of the top plate. This is demonstrated with the large vectors of $-\nabla\dot{\gamma}$ shown there. This is not a positive result, since a

[8] It is clear that a test material with $\eta \to \infty$ (i.e. a rigid body motion) gives no inclination.

large amount of torque is generated at the side of the top plate. In both cases of $h_{\text{cone}} = 16\,\text{cm}$ ($(z_1, z_2) = (16.2, 18.2)\,\text{cm}$) and $h_{\text{cone}} = 8\,\text{cm}$ ($(z_1, z_2) = (8.2, 10.2)\,\text{cm}$), the torque applied from the test material, on the side of the top plate, is about:

$$\hat{\mathbf{T}} = \hat{T}\,\mathbf{i}_{\text{z}} = \int_{z_1}^{z_2}\int_{0}^{2\,\pi} R_{\text{i}}\,\mathbf{i}_{\text{r}} \times (\mathbf{i}_{\text{r}} \cdot \mathbf{T}|_{r=R_{\text{i}}}\,R_{\text{i}}\,d\theta\,dz) \approx 2.6\,\text{Nm}\,\mathbf{i}_{\text{z}} \tag{10.5}$$

and likewise the torque applied on the bottom of the top plate is

$$\hat{\mathbf{T}} = \hat{T}\,\mathbf{i}_{\text{z}} = \int_{0}^{R_{\text{i}}}\int_{0}^{2\,\pi} r\,\mathbf{i}_{\text{r}} \times (-\mathbf{i}_{\text{z}} \cdot \mathbf{T}|_{z=z_1}\,r\,d\theta\,dr) \approx 3.9\,\text{Nm}\,\mathbf{i}_{\text{z}} \tag{10.6}$$

Hence, the total torque applied on the top plate is approximately 6.5 Nm. This means that the torque applied on the side is about 40% of the total torque. As such, the logged torque $\hat{T}$ could be decreasing by about 40% during a measurement. For *stepwise decreasing shear rate sequence*, this would again lead to a torque curve similar to what is shown in Figure 10.18, resulting in a smaller yield value τ_{o}. The effect on the plastic viscosity μ would be unclear (see the illustration on Page 250).

10.3.6 Parallel Plates-Geometry

The solution geometry and boundary condition is represented with the right illustration of Figure 10.25. Figure 10.28, demonstrates the gradient field of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_{\text{c}}$ when the gap between the bottom- and the top plate is 16.2 cm (to the left) and 8.2 cm (to the right). As before, the gap of the latter is reduced by half, using $\Delta z = 1\,\text{mm}$, instead of $\Delta z = 2\,\text{mm}$.

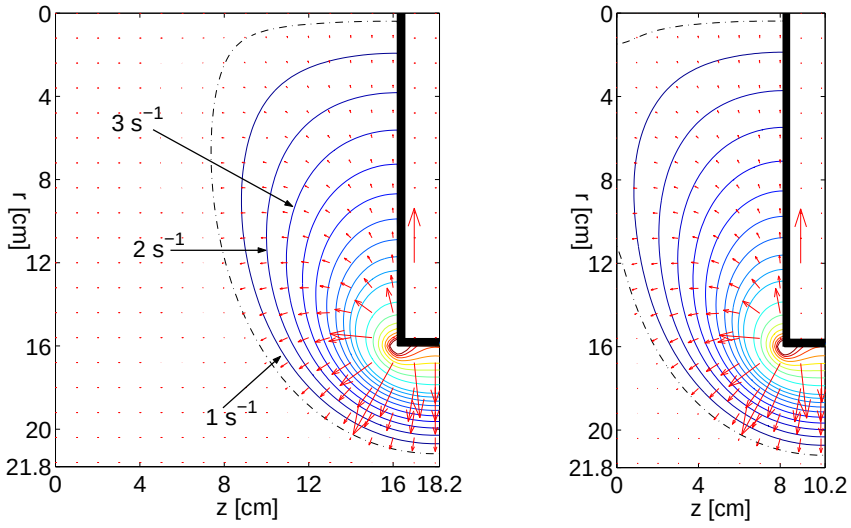


Figure 10.28: Gradient field of shear rate: $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_{\text{c}}$ when the total gap width is 16.2 cm (to the left) and 8.2 cm (to the right). The length of the vector, shown inside the top plate, is $\|\nabla\dot{\gamma}\| = 10^3\,\text{m}^{-1}\text{s}^{-1}$. The isoplot is of shear rate $\dot{\gamma}_{\text{c}}$ and the dashed dotted line shows the boundary between a solid state and a viscoplastic state. $[(\tau_{\text{o}}, \mu) = (200, 20\text{s})\text{Pa};\ \omega_{\text{o}} = 3\text{rad/s};\ \delta = 4 \cdot 10^{-3}\text{s}^{-1};\ (\Delta r, \Delta z) = (2, 2 \wedge 1)\text{mm};\ \Delta t = 10^{-6}\text{s}$ (left) $\wedge\ 0.5 \cdot 10^{-6}\text{s}$ (right)].

Relative to potential gravel migration, the difference between the results produced in Figures 10.27 and 10.28 are not large. Approximately the same directions and

lengths of the vectors $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_c$ are produced for all cases. Also, the amount of torque applied on the top plate is more or less the same as for Figures 10.27 and 10.28; i.e. the torque applied on the side of the top plate is about 2.6 Nm and the total torque applied on the top plate, as a whole, is about 6.5 Nm. The torque is now positive, in contradiction to what is shown in Section 10.3.4. This is because of how the test material is now trying to rotate the stationary top plate (i.e. the measuring unit). More precisely, in Section 10.3.4, the test material was always trying to slow down the rotating top plate and hence a negative torque value resulted there.

The above demonstrates that there is no apparent benefit in using the cone-geometry relative to the parallel plates-geometry. Since it is easier to construct the parallel plates configuration, the latter[9] will be considered further in the next section.

10.3.7 The C³P²-Geometry

The Characteristic Thickness of the Flow D_{flow}

To avoid the type of problem mentioned in Section 3.4.2, namely the problem of barrier restraint of the inner and outer cylinder, it is best to have the characteristic thickness of the flow D_{flow} larger or equal to $8D_{\max}$. With this in mind, a shearing zone of $D_{\text{flow}} = 8.2\,\text{cm}$ (right illustration of Figure 10.28) is not considered to be the best option. For example, using $D_{\max} = 16\,\text{mm}$, the condition $D_{\text{flow}} = 5D_{\max}$ is achieved. With the pre-mentioned cone-geometry, even still worse condition would appear around the top part of the cone, namely $D_{\text{flow}} \approx 0$. Using a shearing zone of 16.2 cm (see the left illustration of Figure 10.28), will require a test sample of $\pi R_o^2 h_{\text{tot}} = \pi\,(21.8\,\text{cm})^2\,18.2\,\text{cm} \approx 30\,\text{liters}$. In keeping the amount of test sample around 20 liters and having the condition of $D_{\text{flow}} = 8D_{\max}$, the shearing zone is rather set to be $D_{\text{flow}} = 12.0\,\text{cm}$.

Solution Geometry and Boundary Condition

The left illustration of Figure 10.29 demonstrates the solution geometry and boundary condition used for the new dimensions. The spacing of grid points in r- and z-direction are uniform and equal to $\Delta r = \Delta z = 2\,\text{mm}$. With this new geometry, the corner of the top plate is replaced with a (part) cone-geometry. This is done in the attempt to reduce the large shear rate values that always appear when a corner is present. As such, the overall viscometric geometry now used, is actually a combination of three types of geometries, the **C**one, the **C**oaxial **C**ylinders and the **P**arallel **P**late type. Therefore its designation given here will be the CCCPP-geometry, or simply the C^3P^2-geometry. Figure 10.30 demonstrates a three dimensional vector plot of velocity $\mathbf{v}$ inside the C^3P^2-geometry. As shown, it consist of a rotating bottom part and a stationary top part.

Shear Rate

The right illustration of Figure 10.29 presents the computed gradient field of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_c$. The replacement of the corner (of the top plate) with a cone-geometry, is somewhat less successful than was hoped for. Comparing the above-mentioned

[9] This decision is also made in the light of the possible and unwanted direct torque transportation from the rotating top part of the cone, to the measuring unit, similar to what is shown in Figure 3.8 (Page 62). As discussed in Section 3.4.2, such process results in a larger measured shear viscosity $\eta \to \eta + \Delta\eta$.

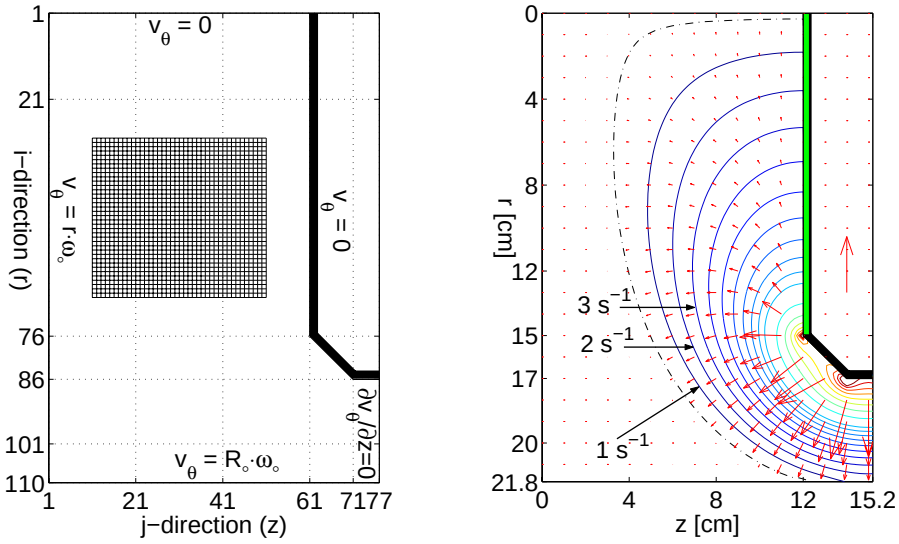


Figure 10.29: To the left: Solution geometry and boundary condition for the C^3P^2-geometry. To the right: Gradient field of shear rate: $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}_c$. The length of the vector, shown inside the top plate, is $\|\nabla\dot{\gamma}\| = 10^3\,\mathrm{m}^{-1}\mathrm{s}^{-1}$. The isoplot is of shear rate $\dot{\gamma}_c$ and the dashed dotted line shows the boundary between a solid state and a viscoplastic state. $[(\tau_o, \mu) = (200, 20\mathrm{s})\mathrm{Pa}$; $\omega_o = 3\mathrm{rad/s}$; $\delta = 2\cdot 10^{-3}\mathrm{s}^{-1}$; $(\Delta r, \Delta z) = (2, 2)\mathrm{mm}$; $\Delta t = 10^{-5}\mathrm{s}]$.

result with the results shown in Figures 10.27 and 10.28 demonstrates only a slight reduction in the shear rate gradient. This means that almost the same large potential for gravel migration is still present. As mentioned before, this potential occurs at the worst location possible, namely where the largest torque (per unit area) is generated (c.f. Equation 10.5).

Change in Velocity Profile

Because of the potential gravel migration around the cone-geometry and near the side of the top plate, the measuring unit will not consist of the whole top plate. Rather, it will only consist a part of it, as shown with the green line in the right illustration of Figure 10.29. This is more apparent with Figure 10.31, where the top plate is divided into a measuring unit and a filter unit. More specifically, the measuring unit consist of the boundary $(\mathrm{i}, \mathrm{j}) = (1 : 76, 61)$. The radius at which the torque is applied, from the test material, is designated with $R_m = (76 - 1)\,\Delta r = 15\,\mathrm{cm}$. This torque is calculated with Equation 10.7.

$$\hat{\mathbf{T}} = \hat{T}\,\mathbf{i}_z = \int_0^{R_m}\int_0^{2\pi} r\,\mathbf{i}_r \times (-\mathbf{i}_z \cdot \mathbf{T}|_{z=12\,\mathrm{cm}}\, r\,d\theta\,dr) \tag{10.7}$$

It should be noted that the above torque-compensation will be only in part effective. This is so because gravel migration will change the velocity profile $v_\theta(r, z)$, beyond what can be predicted by Equation 7.6 (Page 157) alone. Hence, a different torque will be generated at the measuring unit, than predicted with the calculated velocity profile $v_\theta(r, z)$. The difference will occur although no gravel migration is occurring near and at the measuring unit. However, as has been mentioned in Section 10.1,

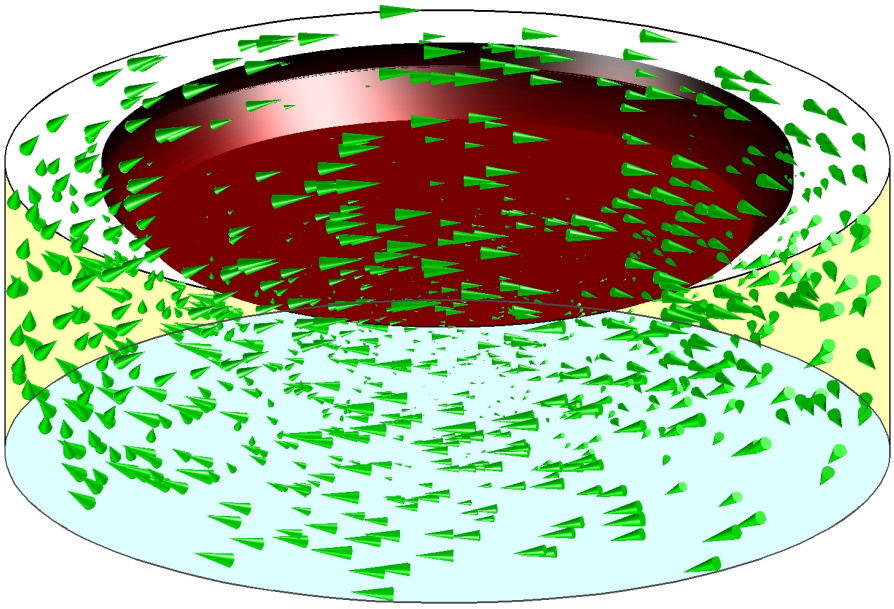


Figure 10.30: Three dimensional vector plot of velocity $\mathbf{v}$, inside the C^3P^2-geometry. $[(\tau_o, \mu) = (200, 20s)Pa$; $\omega_o = 3rad/s$; $\delta = 2 \cdot 10^{-3}s^{-1}$; $(\Delta r, \Delta z) = (2, 2)mm$; $\Delta t = 10^{-5}s]$.

gravel migration $\mathbf{q}$ is not only governed by the gradient in shear rate $-\nabla\dot{\gamma}$, but also by some internal and external properties, described with the migration coefficient ϑ. More precisely, gravel migration can be stated with $\mathbf{q} = -\vartheta\,\nabla\dot{\gamma}$, where the migration coefficient $\vartheta = \vartheta(\Phi, \eta_m, D_{flow}/D_{max}, \dots)$ describes how much amount of gravel should migrate $\mathbf{q}$ for the given difference in shear rate $\nabla\dot{\gamma}$. For the reason mentioned in Section 10.3.8, one could assume that the migration coefficient ϑ is smaller inside the C^3P^2-geometry, compared to (for example) the coaxial cylinders-geometry shown in Figure 10.3. With this in mind, the C^3P^2-geometry is explored further.

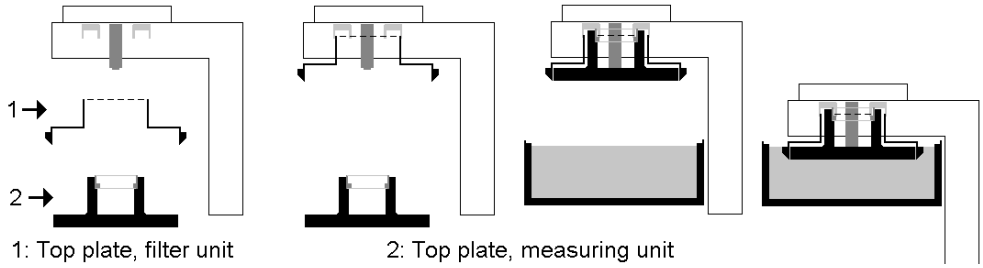


Figure 10.31: The assembly of the C^3P^2-geometry to the ConTec **BML** Viscometer 3.

Applied Torque

The shear stress $\tau = \sqrt{(\mathbf{T} : \mathbf{T})/2}$ is given by $\tau = \eta\,\dot{\gamma} = \mu\,\dot{\gamma} + \tau_o$. Since there is a linear relationship between τ and $\dot{\gamma}$, one could expect that the same would apply for the measured torque $\hat{T}$ and the rotational frequency f_o. In other words, one could expect a relationship of the type $\hat{T} = H\,f_o + G$. As such, it can be assumed that

there is one-to-one correspondence between τ_o and G at the one hand, and between μ and H at the other.

The immediate problem with the C^3P^2-geometry is that there is no analytical approach available in converting the G and H values to yield value τ_o and plastic viscosity μ (like what is shown with Equations 3.26 and 3.27, Page 59). However, because of the pre-mentioned one-to-one correspondence, there is a way around this. It consists of making series of numerical simulations with different sets of viscometric values (τ_o, μ) to produce the different sets of (G, H) values. With this approach, it is possible to produce a table (or a figure) that maps the relationship between the measured G and H values, with the actual viscometric values τ_o and μ of the test sample. An example of such approach is shown in Figure 10.32. The results of this figure will be discussed shortly.

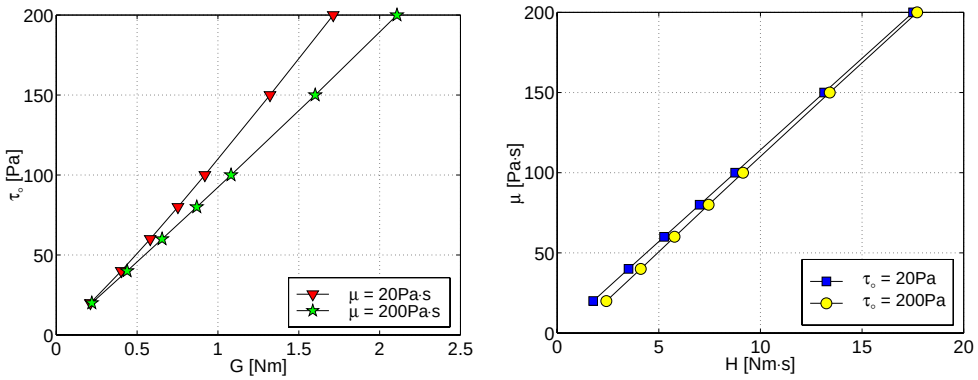


Figure 10.32: Relationship between τ_o and G (to the left) and μ and H (to the right) for the C^3P^2-geometry. Total number of 140 simulations where used in generating the two illustrations.

Effect of Plug on Measured Torque

Figure 10.33 demonstrates the plug propagation inside the C^3P^2-geometry, with decreasing rotational frequency: $f_o = \omega_o/(2\,\pi) = 0.5,\ 0.4,\ 0.3,\ 0.2$ and $0.1\,\text{rps}$. The solid isolines demonstrate the velocity profile. These lines start with the value of $v_\theta = 1\,\text{cm/s}$ near the center of the geometry (i.e. near $r = 0$) and has an equal increment of $\Delta v_\theta = 5\,\text{cm/s}$ towards the outer cylinder (i.e. $v_\theta = [1, 6, 11, \ldots]\,\text{cm/s}$). The dashed dotted isoline is of $\tau_c = \tau_o = 200\,\text{Pa}$ and demonstrates the location of the boundary between the solid state Ω_e and the viscoplastic state Ω_p. As discussed previously, the velocity profile of a pure solid state consists of $v_\theta = r\,\omega_o$. This is produced in the calculations as noted with the even spaces between the isolines present in the solid domain Ω_e. The regularization parameter now consists of $\delta = 2 \cdot 10^{-3}\,\text{s}^{-1}$.

When the *stepwise decreasing shear rate sequence* approach is used, the condition $\dot{\gamma} = 0$ (i.e. solid state) is expected to start at the bottom plate and propagate towards the top plate with decreasing rotational frequency f_o. This result is produced in Figure 10.33 with the movement of the boundary between the solid state Ω_e and the viscoplastic state Ω_p. Because of this, a kind of downward torque profile results, similar to what is shown with the left illustration of Figure 3.10 (Page 66). The bottom-right illustration of Figure 10.33 demonstrates two torque profiles. One of them is marked with circles ○ and demonstrates the simulated result at $f_o = 0.5\,\text{rps}$

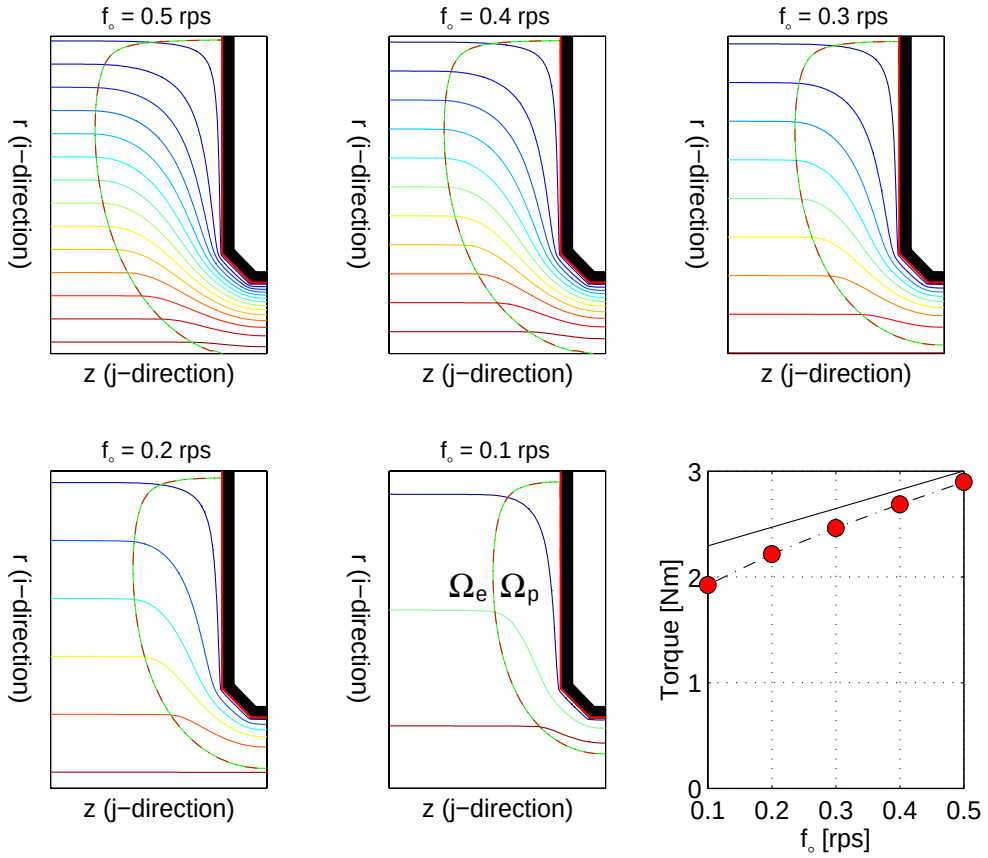


Figure 10.33: Plug propagation inside the C^3P^2-geometry with reducing rotational frequency f_o. Making a linear regression through all the torque points, gives the slope $H = 2.42\,\mathrm{Nm}\cdot\mathrm{s}$ and the point of intersection with the ordinate $G = 1.71\,\mathrm{Nm}$. $[(\tau_o, \mu) = (200, 20\mathrm{s})\mathrm{Pa};\ \delta = 2\cdot 10^{-3}\mathrm{s}^{-1};\ (\Delta r, \Delta z) = (2,2)\mathrm{mm};\ \Delta t = 10^{-5}\mathrm{s}]$.

and down to 0.1 rps. Making a linear regression through these torque points ($\hat{T} = H\,f_o + G$), produces the slope $H = 2.42\,\mathrm{Nm}\cdot\mathrm{s}$ and the point of intersection with the ordinate $G = 1.71\,\mathrm{Nm}$. The torque profile marked with the black solid line is an extrapolation from the result produced in Figure 10.34. The same viscometric values τ_o and μ are applied for this latter case and as such, the same G and H value should be produced in Figures 10.33 and 10.34. But due to plug, this is not the case. With almost no plug occurring for the case in Figure 10.34, different values $H = 1.77\,\mathrm{Nm}\cdot\mathrm{s}$ and $G = 2.12\,\mathrm{Nm}$ are obtained, which are more correct. Their difference is calculated to be: $\Delta H/H = -36.7\%$ and $\Delta G/G = 19.3\%$. This could be considered acceptable, as a worst case scenario. This is so since often the condition $\tau_o/\mu \leq 10\,\mathrm{s}^{-1}$ applies for the test material, which gives a less plug state. This is especially true for the new generation of concrete, namely the self-compacting concrete.

The isolines in Figure 10.34 start with the value of $v_\theta = 1\,\mathrm{cm/s}$ near the center of the geometry (i.e. near $r = 0$) and have an equal increment of $\Delta v_\theta = 25\,\mathrm{cm/s}$ towards the outer cylinder (i.e. $v_\theta = [1, 26, 51, \ldots]\,\mathrm{cm/s}$).

Figure 10.32 demonstrates a mapping, or a relationship between the “measured”

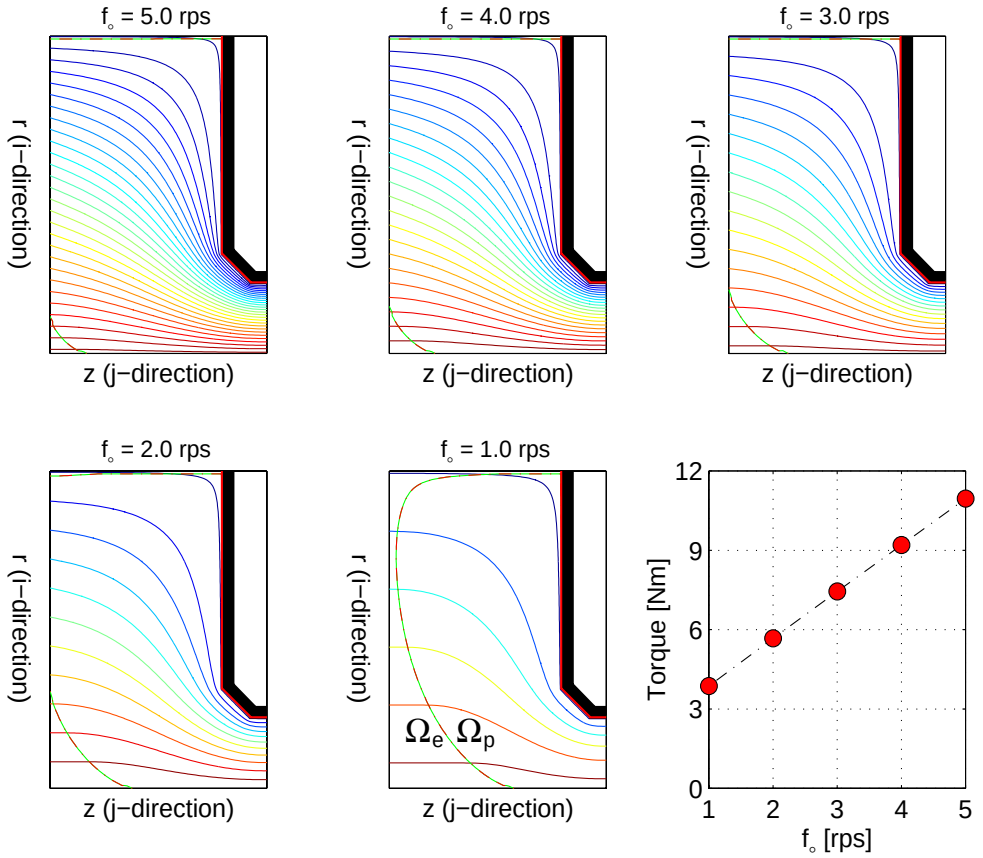


Figure 10.34: Torque profile generated when almost no plug is occuing inside the C^3P^2-geometry with reducing rotational frequency f_o. Making a linear regression through all the torque points, gives the slope $H = 1.77\,\mathrm{Nm}\cdot\mathrm{s}$ and the point of intersection with the ordinate $G = 2.12\,\mathrm{Nm}$. $[(\tau_o, \mu) = (200, 20\mathrm{s})\mathrm{Pa};\ \delta = 2\cdot 10^{-3}\mathrm{s}^{-1};\ (\Delta r, \Delta z) = (2,2)\mathrm{mm};\ \Delta t = 10^{-5}\mathrm{s}]$.

G and H values, with the actual viscometric values τ_o and μ used in the numerical simulations. The two lines shown in each illustration do not coincide because of how plug influences the calculation of G and H values to the different degree. However, with these two illustrations it is possible to obtain τ_o and μ, from measured G and H values. For example, measuring $H \approx 2.3\,\mathrm{Nm}\cdot\mathrm{s}$ gives $\mu \approx 20\,\mathrm{Pa}\cdot\mathrm{s}$ from the right illustration of Figure 10.32. Then with measured $G \approx 1.3\,\mathrm{Nm}$, the left illustration gives $\tau_o \approx 150\,\mathrm{Pa}$ (extracted by using the "$\mu = 20\,\mathrm{Pa}\cdot\mathrm{s}$"-line). Obviously, more simulations are needed to gain a more complete map between (τ_o,μ) and (G,H).

10.3.8 Summary and Conclusion (Part II)

As has been mentioned in this chapter, gravel migration $\mathbf{q}$ (or particle migration in general) is not only governed by the gradient in shear rate $-\nabla\dot{\gamma}$ alone, but also by some internal and external properties, described with the migration coefficient ϑ. More precisely, gravel migration can be stated with $\mathbf{q} = -\vartheta\,\nabla\dot{\gamma}$, where the migration coefficient $\vartheta = \vartheta(\Phi, \eta_m, D_{flow}/D_{max}, \dots)$ describes how much amount of gravel should

migrate $\mathbf{q}$ for the given difference in shear rate $\nabla\dot{\gamma}$. In Sections 10.1 and 10.3.1, gravel migration is related to three different physical processes. These are summarized in the following box. All these processes are included in the (total) migration coefficient ϑ, which is suggested here, to be of the functional form $\vartheta = \vartheta_{\mathrm{bg}} + \vartheta_{\mathrm{dt}} + \vartheta_{\mathrm{cf}}$.

Possible reasons for particle migration ($\boldsymbol{\vartheta = \vartheta_{\mathrm{bg}} + \vartheta_{\mathrm{dt}} + \vartheta_{\mathrm{cf}}}$).

1. **Background Effect ($\boldsymbol{\vartheta_{\mathrm{bg}}}$):** The background effect is related to the difference in collision rate $-\nabla\dot{n}$ by freely moving particles (given with the condition $D_{\mathrm{flow}}/D_{\max} \geq 8$, c.f. Section 3.4.2). With the difference in collision rate $-\nabla\dot{n}$ that each suspended particle "feels", they are pushed (by collisions) away from the region of highest collision rate, namely into the direction of $-\nabla\dot{\gamma} \propto -\nabla\dot{n}$.
2. **Dilatancy Effect ($\boldsymbol{\vartheta_{\mathrm{dt}}}$):** The second process is related to a dilatancy effect of densely packed suspended particles (large phase volume Φ). In such suspension the particle distribution Φ must change to permit the suspended particles to flow past one another (see Figure 3.3, Page 54). This means a withdrawal (or suction) of matrix, from the region of smallest deformation (or smallest $\dot{\gamma}$) into region of largest deformation (or largest $\dot{\gamma}$); i.e. the matrix must move in the direction of $\nabla\dot{\gamma}$. With a (observed) constant volume of the overall suspension (inside the viscometer), the suspended particles must move in the opposite direction at the same time, namely in the direction of $-\nabla\dot{\gamma}$.
3. **Confinement Effect ($\boldsymbol{\vartheta_{\mathrm{cf}}}$):** The gravel migration shown in Figure 10.2, could be more as a result of the relative small gap system of $(R_{\mathrm{i}}, R_{\mathrm{o}}) = (10, 14.5)$ cm, giving the condition of $D_{\mathrm{flow}}/D_{\max} \approx 3$. As mentioned in Section 3.4.2, such condition results in a harder and more frequent collisions between the larger aggregate particles. This could result in a stronger "pushing mechanism" than anticipated in Item 1 above. In this case, the suspended particles are assumed to be pushed in the direction of $-\nabla\dot{\gamma}$, just as applies in Item 1.

In Sections 10.3.4, 10.3.5, 10.3.6, and 10.3.7, the possibility for a different geometry is explored for the ConTec **BML** Viscometer 3. This was done in the anticipation of solving the problem of gravel migration. The analysis is based on the gradient of shear rate $\mathbf{q}/\vartheta = -\nabla\dot{\gamma}$, where the geometry that has the lowest difference in shear rate (i.e. has the has smallest vectors), is considered to be the best solution. Since no geometry gave a real reduction in shear rate gradient, the overall investigations became somewhat less successful in that respect than was hoped for. However, in the above box, three different processes are related to the gravel migration. Of the three, the last one could be the most dominating for what is shown in Figure 10.2, i.e. $\vartheta_{\mathrm{cf}} \gg \vartheta_{\mathrm{bg}} + \vartheta_{\mathrm{dt}}$ meaning $\vartheta \approx \vartheta_{\mathrm{cf}}$. With the condition of $D_{\mathrm{flow}}/D_{\max} = 8$, the modification shown in Section 10.3.7 might be more successful than indicated with the right illustration of Figure 10.29. In relation to this, increasing the current geometry of the ConTec **BML** Viscometer 3 could also be very much beneficial, however would require a larger test sample. Such maneuver is discussed in Section 10.3.1.

In Section 10.3.4, the BTRHEOM-geometry was considered for the ConTec **BML** Viscometer 3. Since the difference between the full-slippage-case and the no-slippage-case is very large relative to the potential gravel migration, **and also** relative to the applied torque, there are some serious uncertainties related to this geometry. Because of these two uncertainties, the BTRHEOM-geometry is not considered to be better than the other parallel plates-geometries, investigated in this thesis.

The summary of the experimental part of this chapter is in Section 10.2.3.

10.3.9 Recommended Future Research

In this thesis, gravel migration was only investigated on "classical" concretes, with slump about 200 mm and below. It would be very interesting to investigate this phenomenon on the new type of concrete, namely the self-compacting concrete. There, the ratio of τ_o/μ would be much lower than present in the current work and hence, gravel migration should be less present. Testing this hypothesis would be a good next step.

With the large shearing zone of the C^3P^2-geometry, it would be very interesting to construct this viscometric geometry and then using it on the ConTec **BML** Viscometer 3 as demonstrated in Figure 10.31. As shown with Figure 10.32, a relationship between the measured G and H values and the actual viscometric values τ_o and μ can be established, which makes this geometry quite usable. The interested reader can create a more extensive relationship between (G, H) and (τ_o, μ) as the source code for the C^3P^2-geometry is available in Appendix A.

Chapter 11

Summary and Final Remarks

11.1 Summaries and Conclusions (Parts I to III)

Three major subjects are considered in this thesis, and consequently it is divided into three parts. The **first part** concerns how the different lignosulfonate types changes the rheological properties of the cement based material as a function of temperature and time (Section 1.2.1). This investigation is made with help from the **second part** of this thesis, which identifies some of the parameters affecting the shear viscosity $\eta = \eta(J_t, J_t^P, \dots)$ of the cement based material. This is done by investigating the thixotropic behavior of cement paste (Section 1.2.2). An experimental error is present during a viscometric measurement on concrete. The error is generated by gravel migration. Investigating and compensating for this error constitutes the **third part** of this thesis (Section 1.2.3).

Given the overall structure of this thesis, a **summary and conclusion** is provided at the end of each part. More precisely, the summary and conclusion for the **first part** (the effects of lignosulfonates) is given in Section 6.5 (Page 152). The summary and conclusion for the **second part** (thixotropy) is given in Section 9.9 (Page 233). Finally, for the issue of gravel migration (the **third part**), the summary and conclusion is given in Section 10.2.3 (the experimental part) and in Section 10.3.8 (the numerical part), on Pages 249 and 267, respectively. Recommended future research is also given at the end of each part of this thesis.

In the following, the most important conclusions for each part are given:

Conclusions of the first part: Effects of lignosulfonates (Section 6.5).

1. The rheological effects of the lignosulfonates are basically determined by their molecular weights. The largest molecular weight product gives the best result in terms of workability and workability retention.
2. For the high molecular weight lignosulfonates (**HMW** and **VHMW**), the steric potential energy V_S is more or less overshadowing the electrostatic potential energy V_R (in the process of dispersing the cement particles). For such a case, issues like ionic strength I or zeta potential ζ becomes less important.
3. For the low molecular weight lignosulfonates (**LMW**), the steric potential V_S is lower, allowing the electrostatic potential energy V_R to become more important.
4. The effects of the different lignosulfonate types are primarily reflected in the measured yield value τ_o. The measured plastic viscosity μ changes little.

Conclusions of the second part: Thixotropy (Section 9.9).

5. Thixotropy is mostly governed by a combination of reversible coagulation, dispersion and then re-coagulation of the cement particles.
6. In numerical simulation of the thixotropic behavior of the cement paste, it was necessary to introduce a fading memory of shear rate $\dot{\gamma}$ and of coagulation rate H, into the shear viscosity equation η (Section 9.3.1).
7. Also, two types of yield values had to be introduced into the simulation, namely τ_o and $\tilde{\tau}_o$. The former (τ_o) is related to the permanent coagulation state J_t^p of the cement particles, while the latter ($\tilde{\tau}_o$) is related to the reversible coagulation state J_t (see Figure 2.11, Page 27). The same type of relationship applies for the plastic viscosity μ and its thixotropic counterpart $\tilde{\mu}$.

Conclusions of the third part: Gravel migration (Section 10.2.3).

8. The measured viscometric values of concrete are somewhat incorrect due to gravel migration.
9. Usually, gravel migration was finished after about 20 to 30 seconds after start of rotation of the outer cylinder R_o. [Elsewhere, this time duration could be longer or shorter, depending on type of test material, viscometer, angular velocity and so forth].
10. The best way to process the logged torque data, is to dismiss the first torque points $\hat{T}$, when gravel concentration $\Phi = \Phi(r, z, t)$ (i.e. phase volume) is changing most rapidly. In taking this approach, one is only extracting the viscometric values related to the remaining fat concrete that surrounds the inner cylinder (R_i), and not of the concrete in a homogenous state.
11. However, in Section 6.3, a good relationship between the Abrams slump and the viscometer is established, indicating that the fat concrete simulates well the behavior of the concrete in a homogeneous state. Hence, regardless of gravel migration, concrete viscometers are still of large relevance, although they do not provide correct absolute values for such cases.
12. Gravel migration does not necessarily have to occur. In a different project concerning self-compacting concrete, gravel migration was much less present and often not observed at all. The same viscometer was used as in this thesis.

11.2 Experimental Errors of Different Types

Different types of errors have been considered, which affect the measured rheological values in the **first part** of this thesis. For the concrete case, the error of gravel migration has been present (Chapter 10), while for the mortar case, the issues of plug flow (Section 3.5.3) and air entrainment (Section 5.5.3) have been the main problems. Investigating these errors has been rather time consuming, and frequently it has been tempting to ignore these errors altogether, and dive directly into the main investigation at hand. That is, to investigate the effects of the different lignosulfonates on the rheological behavior of the cement based material.

Compensating for One Error and Making Room for Another

Compensating for one experimental error can easily result in a new and even worse error. For example, if the problem of plug flow is present, one might be tempted to increase the maximum value of angular velocity $\omega_{max} = 2\,\pi\,f_{max}$ to compensate

for this (see Table 3.1, Page 64). However, such a maneuver could result in particle migration (see discussion in Section 3.5.1). The latter error is a worse type of error because of its unknown magnitude. In Section 3.5.3, the magnitude of the error produced by plug flow is successfully estimated. This does not apply for the error produced by particle migration in Chapter 10. There, only the viscometric values of the fat concrete near the inner cylinder, is obtained. The viscometric values of the fresh concrete in a homogenous state is not available. Hence, the magnitude of the gravel migration error cannot be estimated.

Segregation and Stability

None of the batches in this thesis segregated under manual agitation, since they were designed in such manner. Therefore, the effect of normal (vertical) segregation is not of concern in this work. With this degree of stability, all the batches could at least tolerate an acceleration of $g = 9.81\text{m/s}^2$. As shown with the simple example on Page 240, gravitational acceleration g is about ten times larger than centripetal acceleration. Therefore, centripetal acceleration could not be responsible for the observed horizontal segregation (i.e. particle migration).

Particle Migration and the Herschel-Bulkley Model

Generally, particle migration can be present for any viscometer used for testing particle suspensions. When occurring, a different torque curve $\hat{T}$ results, compared to what would be measured for the suspension in a homogeneous state. Hence, such particle migration will influence the viscometric results. For example, a non-linear torque $\hat{T}$ generated by particle migration, could incorrectly lead to the suggestion that the Herschel-Bulkley model applies $\eta = \mu\,\dot{\gamma}^{n-1} + \tau_o/\dot{\gamma}$, when in fact the tested material corresponds to Bingham model $\eta = \mu + \tau_o/\dot{\gamma}$ (when in a homogeneous state).

Particle Migration, Thixotropy and Hysteresis Loop

Thixotropy is frequently investigated by measuring the torque $\hat{T}$ under a linear increase and then decrease in angular velocity ω_o. The casual perception is that if the two torque curves produced do not coincide, but form a hysteresis loop, the tested sample is thixotropic (the area of a loop is then used as a measure of the degree of thixotropy). It is very important to note that such hysteresis loop can also be caused by particle migration during the experiment. If thixotropic behavior and particle migration are occurring simultaneously, separating the two would be a challenging task. One could wait a while until particle migration is finished and thereafter measure (then diminished) thixotropic behavior.

In this thesis, the duration of particle migration was 20 to 30 seconds. In few cases (see Figures 10.7 and 10.11, Pages 243 and 246), some thixotropic behavior was observed after this time and was considered to be thixotropy of the remaining fat concrete that surrounds the inner cylinder (R_i).

Particle Migration versus Plug Flow

Generally, particle migration can be present for any viscometer used for testing particle suspensions. When occurring, the resulting torque curve could easily be mistaken for a thixotropic property of the tested material. When making a thixotropic analysis, it must be ascertained that the test material is homogenous before and after a

measurement. If particle migration is occurring, a reduction in maximum value of angular velocity $\omega_{\max} = 2\,\pi\,f_{\max}$ must be made (see Table 3.1, Page 64), with the constraint that the corresponding shear rate $\dot{\gamma}$ is still of relevance, c.f. Section 4.4.1. Such maneuver could however result in plug flow. This is sometimes the case in Chapter 9, but there the plug flow is not a problem, since it is included and accounted for in the numerical simulations. The experience of this thesis is that plug flow is more easily treated and compensated for than particle migration.

11.3 The Relevance of Viscometers

In Chapter 6, the change in shear viscosity $\eta = \mu + \tau_o/\dot{\gamma}$ as a function of time and temperature, reflects changes in the yield value τ_o, rather than in the plastic viscosity μ. That is, the yield value τ_o is more sensitive to temperature and time, compared to the plastic viscosity μ (see also the numbered items on Page 126). Also, the different lignosulfonate types influence the yield value τ_o much more than the plastic viscosity μ. That is, comparing the effect of two different lignosulfonate types gives usually a different yield value τ_o, while the plastic viscosity μ is roughly the same. Therefore, for the mixes tested in Chapter 6, the characteristics can be described with only a single parameter, namely with the yield value τ_o. In Section 6.3.1, a strong relationship between yield value τ_o and slump value S was established. This indicates that the whole research program on lignosulfonates could have been carried out using only slump measurements. Of course, such a statement does not have general validity for the following reasons:

- That the lignosulfonates are more affecting the yield value τ_o compared to the plastic viscosity μ, is an important result by itself. In fact, such result has been reported elsewhere when using lignosulfonates [128]. This type of result could never been gained by slump measurements alone.
- When comparing the plastic viscosity μ between the different batch types, different values result. For example, the OPC-batches of $w/c = 0.5$ (Figure 6.9, Page 112) gives $\mu \approx 1\,\text{Pa}\cdot\text{s}$, while the FAC-batches of $w/c = 0.57$ (Figure 6.19, Page 120) gives $\mu \approx 4\,\text{Pa}\cdot\text{s}$.
- For the past 25 years, it has been demonstrated that fresh concrete, mortar and cement paste can, with good accuracy, be considered as Bingham materials [28]. That is, the cement based materials can successfully be described with the yield value τ_o and plastic viscosity μ. With small and roughly constant μ, the present concrete and mortar mixes can be characterized by τ_o alone (see Chapter 6). This is analogous to a Newtonian liquid, which is characterized by μ alone (τ_o is zero). Thus for a given type of concrete, one parameter may be sufficient, but this can only be ascertained by viscometric tests.
- The type of rheological investigations done in Chapter 9, could never been made with the simple slump test. This underlines the importance of viscometers above the more traditional slump measuring devise.
- In general, the plastic viscosity μ is more changeable than seen for the mixes in this thesis. For example, in certain cases of self-compacting concrete, the plastic viscosity μ plays an extremely important role. There, the yield value τ_o is very low, while the rheological characteristic of the concrete is more given by the plastic viscosity μ.

Bibliography

[1] D. A. Abrams. Design of concrete mixes. In *Structural Materials Research Laboratory*. Levis Institute, 1925. Reprints from Minutes of the Annual Meeting of the Portland Cement Association, New York, December 1918.

[2] J. D. Anderson. *Computational Fluid Dynamics, The Basics with Applications*. McGraw-Hill, Inc., USA, 1995.

[3] T. T. Arny. *Explorations, an Introduction to Astronomy*. Mosby-Year Book, Inc., USA, 1994.

[4] R. A. Bagnold. Experiments on a gravity-free dispersion of large solid spheres in a newtonian fluid under shear. In *Proc. Roy. Soc. Lond.*, volume A 225, London, 1954.

[5] P. F. G. Banfill. Experimental investigations of the rheology of fresh mortar. In *Properties of Fresh Concrete, Proceedings of the RILEM Colloquium*, Great Britain, 1990. University Press, Cambridge.

[6] P. F. G. Banfill. The rheology of fresh mortar. *Magazine of Concrete Research*, 43(154):13–21, 1991.

[7] P. F. G. Banfill. Structural breakdown and the rheology of cement mortar. In P. Moldenaers and R. Keunings, editors, *Theoretical and Applied Rheology; Proc. XI^{th} Int. Congr. on Rheology*, Brussels, Belgium, 1992. Elsevier Science Publishers B. V.

[8] P. F. G. Banfill, D. Beaupré, F. Chapdelaine, F. de Larrard, P. Domone, L. Nachbaur, T. Sedran, O. H. Wallevik, and J. E. Wallevik. *Comparison of Concrete Rheometers: International Tests at LCPC (Nantes, France) in October, 2000 (NISTIR 6819)*. National Institute of Standard and Technology (NIST), Gaithersburg, USA, 2001. Editors: F. Ferraris and L. E. Brower.

[9] H. A. Barnes, J. F. Hutton, and K. Walters. *An Introduction to Rheology*. Elsevier Science B. V., Netherlands, 1989.

[10] P. J. M. Bartos, M. Sonebi, and A. K. Tamimi (editors). *Workability and Rheology of Fresh Concrete: Compendium of Tests; Report of RILEM Technical Committee TC 145-WSM, Workability of Special Concrete Mixes*. RILEM Publications S.A.R.L., Cachan Cedex, France, 2002.

[11] E. Battaner. *Astrophysical Fluid Dynamics*. Cambridge University Press, Great Britain, 1996.

[12] A. Beiser. *Concepts of Modern Physics*. McGraw-Hill Book Co., Singapore, 1987.

[13] M. Bercovier and Engelman M. A finite element method for incompressible non-Newtonian flows. *Journal of Computational Physics*, 36:313–326, 1980.

[14] J. D. Bernal, J. W. Jeffery, and H. F. W. Taylor. Crystallographic research on the hydration of portland cement: A first report on investigation in progress. *Magazine of Concrete research*, 11:49–54, 1952.

[15] R. B. Bird, R. C. Armstrong, and O. Hassager. *Dynamics of Polymeric Liquids*. John Wiley & Sons, Inc., USA, 2nd edition, 1987.

[16] R. G. Blezard. The history of calcareous cements. In P. C. Hewlett, editor, *Lea's Chemistry of Cement and Concrete*. Arnold (a member of the Hodder Headline Group), Great Britain, 1998.

[17] S. J. Bloomer. *Further Development of the Two-point Test for the Measurement of the Workability of Concrete*. PhD thesis, University of Sheffield, Sheffield, Great Britain, 1979.

[18] R & D Borregaard LignoTech Inc. URL *http://www.borregaard.com*.

[19] R & D Borregaard LignoTech Inc. Borregaard Concrete Handbook. Sarpsborg, Norway, 2001.

[20] R. A. Brown. *Fluid Mechanics of the Atmosphere*. Academic Press, Inc., USA, 1991.

[21] R. L. Burden and J. D. Faires. *Numerical Analysis*. PWS-KENT Publishing Company, Boston,USA, 4th edition, 1988.

[22] C. M. Chapman. Method and apparatus for determining consistency. In *Proceedings ASTM*, volume 13, 1913.

[23] D. K. Cheng. *Field and Wave Electromagnetics*. Addison Wesley Publishing Company, USA, 2nd edition, 1989.

[24] *Concrete Manual, Concrete Quality and Field Practices*. ICBO, 5360 Workman Mill Road, Whittier, CA 90601-2298, USA, 5th edition, 2000.

[25] F. de Larrard. *Concrete Mixture Proportioning, a Scientific Approach*. F & FN Spon, New York, USA, 1999.

[26] F. de Larrard, C. F. Ferraris, and T. Sedran. Fresh concrete: a Herschel-Bulkley material. *Technical note, Materials and Structures*, 31:494–498, 1998.

[27] F. de Larrard, J. C. Szitkar, C. Hu, and M. Joly. Design of a rheometer for fluid concretes. In *International RILEM Workshop on Special Concretes: Workability and Mixing*, University of Paisley, Scotland, 1993.

[28] P. L. J. Domone, X Yongmo, and P. F. G. Banfill. Developments of the two-point workability test for high-performance concrete. *Magazine of Concrete Research*, 51(3):171–179, 1999.

[29] R. M. Edmeades and P. C. Hewlett. Cement admixtures. In P. C. Hewlett, editor, *Lea's Chemistry of Cement and Concrete*. Arnold (a member of the Hodder Headline Group), Great Britain, 1998.

[30] C. H. Edwards, Jr and D. E. Penney. *Calculus and Analytical Geometry*. Prentice-Hall, Inc., USA, 2nd edition, 1986.

[31] C. F. Ferraris and F. de Larrard. *Testing and Modelling of Fresh Concrete Rheology (NISTIR 6094)*. National Institute of Standard and Technology (NIST), Gaithersburg, USA, 1998.

[32] R. J. Flatt. *Interparticle Forces and Superplasticizers in Cement Suspensions (N° 2040)*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, EPFL, 1999.

[33] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics*, volume I. Springer Series in Computational Physics, Springer-Verlag, Germany, 2nd edition, 1990.

[34] Software Manual for ConTec Viscometers. URL *http://www.contec.is*.

[35] E. M. Gartner, J. F. Young, D. A. Damidot, and I. Jawed. Hydration of portland cement. In J. Bensted and P. Barnes, editors, *Structure and Performance of Cements*. Spon Press, London, Great Britain, 2nd edition, 2002.

[36] P. M. Gerhart, R. J. Cross, and J. I. Hochstein. *Fundamentals of Fluid Mechanics*. Addison Wesley Publishing Company, USA, 1992.

[37] O. Graf. Experiments of the behaviour of reinforcement in concrete of various compositions. *Deutscher Ausschuss für Eisenbeton*, 71:37–60, 1933.

[38] J. Gustafsson. Polymer adsorption and lignosulfonates. In *Borregaard Symposium on Workability and Workability Retention*, Sarpsborg, Norway, 1998.

[39] J. Gustafsson. The dispersing and retarding functions of lignosulfonates. In *Borregaard Symposium on Retardation*, Løkken, Norway, 2000.

[40] U. Håkansson. *Rheology of Fresh Cement-Based Grouts*. PhD thesis, The Royal Institute of Technology, Stockholm, Sweden, 1993.

[41] K. Hattori. Electrokinetics and rheological behavior of cement suspensions. Department of Civil Engineering, Chuo University, Tokyo, Japan.

[42] K. Hattori and K. Izumi. A new viscosity equation for non-newtonian suspensions and its application. In P. F. G. Banfill, editor, *Rheology of Fresh Cement and Concrete. Proceedings of the International Conference Organized by The British Society Of Rheology, University of Liverpool, March 16-29 1990*, London, Great Britain, 1991. E & FN Spon.

[43] E. Haug and H. P. Langtangen. Basic equations in eulerian continuum mechanics. In M. Dæhlen and A. Tveito, editors, *Numerical Methods and Software Tools in Industrial Mathematics*. Birkhauser, Boston, USA, 1997.

[44] S. Helland. *Norwegian Standards dealing with "Curing Technology"*. IPACS, 1999. Project Memo 28.04.99 (IPACS).

[45] W. D. Hibler, III. A viscous sea ice law as a stochastic average of plasticity. *Journal of Geopysical Research*, 82(27):3933–3934, 1977.

[46] W. D. Hibler, III. A dynamic thermodynamic sea ice model. *Journal of Physical Oceanography*, 9(4):817–822, 1979.

[47] J. P. Holman. *Heat Transfer*. McGraw-Hill, Inc., USA, 7th edition, 1990.

[48] J. R. Holton. *An Introduction to Dynamic Meteorology*. Academic Press Inc., California, USA, 3rd edition, 1992.

[49] J. R. Howell and R. O. Buckius. *Fundamentals of Engineering Thermodynamics*. McGraw-Hill, Inc., Singapore, 1987.

[50] C. Hu. *Rhéologie des Bétons Fluides*. PhD thesis of ENPC. Études et Recherches des LPC. Série ouvrages d'art n° 16, La France, 1995.

[51] C. Hu, F. de Larrard, T. Sedran, C. Boulay, F. Bosc, and F. Deflorenne. Validation of BTRHEOM, the new rheometer for soft-to-fluid concrete. *Materials and Structures*, 29(194):620–631, 1996.

[52] R. J. Hunter. *Foundations of Colloid Science*. Oxford University Press, New York, USA, 2nd edition, 2001.

[53] S. C. Hunter. *Mechanics of Continuous Media*. Ellis Horwood Limited, Great Britain, 2nd edition, 1983.

[54] Time-Life Books Inc. Arcata Graphics, Kingsport, Tennessee, USA, 1988.

[55] F. Irgens. *Rheology and non-Newtonian Fluids*. The Norwegian University of Science and Technology, Department of Applied Mechanics, Thermodynamics and Fluid Dynamics, Trondheim, Norway, 1998. (in Norwegian).

[56] J. Israelachvili. *Intermolecular & Surface Forces*. Academic Press, California, USA, 2nd edition, 1991.

[57] T. E. Jakobsson. (Icelandic Meteorological Office). The photo is taken on the research vessel *Bjarni Sæmundsson*.

[58] I. Jawed, J. Skalny, and J. F. Young. Hydration of portland cement. In P. Barnes, editor, *Structure and Performance of Cements*. Applied Science Publishers Limited, Essex, England, 1983.

[59] J. T. Jenkins and S. B. Savage. Theory for the rapid flow of identical, smooth, nearly elastic, spherical particles. *Journal of Fluid Mechanics*, 130:187–202, 1983.

[60] NASA Jet Propulsion Laboratory (JPL). URL *http://www.jpl.nasa.gov*.

[61] H. Justnes. *State-of-the Art on the Induction Period of Cement Paste and the Interaction between Cement and Lignoshulponate [STF22 F97876]*. SINTEF, Civil and Environmental Engineering (Cement and Concrete), Trondheim, Norway, 1997.

[62] H. Justnes. Interaction of lignosulphonates with setting and early hydration of portland cement. In *Borregaard Symposium on Retardation*, Løkken, Norway, 2000.

[63] H. Justnes, E. J. Sellevold, and G. Lundevall. High strength concrete binder. In *Proceedings of Canmet/ACI*, Istanbul, 1992.

[64] H. Justnes, E. J. Sellevold, B. Reyniers, D. Van Loo, A. Van Gemert, F. Verboven, and D. Van Gemert. The influence of cement characteristics on chemical shrinkage. In E. Tazawa, editor, *Autogenous Shrinkage of Concrete*. F & FN Spon, Great Britain, 1999.

[65] H. Justnes, A. Van Gemert, F. Verboven, and E. J. Sellevold. Total and external chemical shrinkage of low w/c ratio cement pastes. *Advances in Cement Research*, 8(31):121–126, 1996.

[66] H. P. Langtangen. *Computational Partial Differential Equations, Numerical Methods and Diffpack Programming*. Lecture Notes in Computational Science and Engineering, Springer-Verlag, Germany, 1999.

[67] R. Lapasin, A. Papo, and S. Rajgelj. The phenomenological description of the thixotropic behavior of fresh cement pastes. *Rheological Acta*, 22:410–416, 1983.

[68] C. D. Lawrence. Physicochemical and mechanical properties of portland cements. In P. C. Hewlett, editor, *Lea's Chemistry of Cement and Concrete*. Arnold (a member of the Hodder Headline Group), Great Britain, 1998.

[69] J. C. Le Bell. *The Influence of Lignosulfonate on the Colloidal Stability of Particulate Suspensions*. PhD thesis, Åbo Akademi, Åbo, Finland, 1983.

[70] D. Leighton and A. Acrivos. The shear-induced migration of particles in concentrated suspensions. *Journal of Fluid Mechanics*, 181:415–439, 1987.

[71] C. C. Lin and L.A. Segel. *Mathematics Applied to Deterministic Problems in the Natral Sciences*. SIAM, Phildadelphia, USA, 1988.

[72] L. E. Malvern. *Introduction to the Mechanics of Continuous Medium*. Prentice-Hall, Inc., USA, 1969.

[73] J. B. Marion and S. T. Thornton. *Classical Dynamics of Particles & Systems*. Harcourt Brace Jovanovich, Inc., USA, 3rd edition, 1988.

[74] G. E. Mase. *Schaums Outline Series: Theory and Problems of Continuum Mechanics*. McGraw-Hill Inc., USA, 1970.

[75] J. Mewis. Thixotropy - a general review. *Journal of Non-Newtonian Fluid Mechanics*, 6:1–20, 1979.

[76] N. B. Milestone. Hydration of tricalcium silicate in the presence of lignosulfonates, glucose, and sodium gluconate. *Journal of the American Ceramic Society*, 62(7-8):321–324, 1979.

[77] A. R. Mitchell and D. F. Griffiths. *The Finite Difference Method in Partial Differential Equations*. John Wiley & Sons Limited, Great Britain, 1980.

[78] J. H. Mork. *The Rheology of Fresh Concrete; Its Relationship to the Relative Content of Cement, Gypsum and Hemihydrate*. PhD thesis, The Norwegian University of Science and Technology (Previously: The Norwegian Institute of Technology), Trondheim, Norway, 1994. Dr.ing. thesis no. 1994:04 (in Norwegian).

[79] P. C. Mørk. *Surface and Colloid Chemistry*. The Norwegian University of Science and Technology, Department of Chemical Engineering, Trondheim, Norway, 5$^{\text{th}}$ edition, 1997. (in Norwegian).

[80] E. Mørtsell. *The Effect of the Constituent Materials on the Rheology of Fresh Concrete*. PhD thesis, The Norwegian University of Science and Technology (Previously: The Norwegian Institute of Technology), Trondheim, Norway, 1996. Dr.ing. thesis (in Norwegian).

[81] E. Mørtsell, M. Maage, and S. Smeplass. A particle-matrix model for the prediction of workability of concrete. In *RILEM International Conference on Production Methods and Workability of Concrete*, Glasgow, Scotland, 1996.

[82] J. Murata and H. Kikukawa. Studies on rheological analysis of fresh concrete. In *Fresh Concrete: Important Properties and their Measurement, Proceedings of a RILEM Seminar held 22-24 March 1973*, University of Leeds, Great Britain, 1973.

[83] J. Murata and H. Kikukawa. Viscosity equation for fresh concrete. *ACI Materials Journal*, 89(3):230–237, 1992.

[84] B. O. Myrvold. Structure and properties of lignosulfonates. In *Borregaard Symposium on Retardation*, Løkken, Norway, 2000.

[85] B. O. Myrvold, 2002. Private communication to Dr. Bernt O. Myrvold (physical chemist), currently employed at Borregaard LignoTech R & D, Norway.

[86] P. Nepper-Christensen. Hvad er beton? In A. D. Herholdt, C. F. P. Justesen, P. Nepper-Christensen, and A. Nielsen, editors, *Beton-Bogen, 2. udgave*. Aalborg Portland, Denmark, 1985.

[87] A. M. Neville. Chairman's summary. In *Fresh Concrete: Important Properties and their Measurement, Proceedings of a RILEM Seminar held 22-24 March 1973*, University of Leeds, Great Britain, 1973.

[88] A. M. Neville. *Properties of Concrete*. Addison Wesley Longman Limited, Great Britain, 1995.

[89] A. M. Neville and J. J. Brooks. *Concrete Technology*. Addison Wesley Longman Limited, Great Britain, 1990.

[90] J. W. Nicholson. *The Chemistry of Polymers*. The Royal Society of Chemistry, Cambridge, Great Britain, 2$^{\text{nd}}$ edition, 1997.

[91] I. Odler. Hydration, setting and hardening of portland cement. In P. C. Hewlett, editor, *Lea's Chemistry of Cement and Concrete*. Arnold (a member of the Hodder Headline Group), Great Britain, 1998.

[92] H. C. Ohanian. *Ohanian Physics*. W. W. Norton & Company, Inc., New York, USA, 2nd edition, 1989.

[93] R. L. Panton. *Inmompressible flow*. John Wiley & Sons, Inc., Singapore, 1984.

[94] A. Papo. Rheological models for cement pastes. *Materials and Structures*, 21, 1988.

[95] B. G. Petersen. *Water Reducing Admixtures for Cement and Mortar Based on Combinations of Lignosulphonates and Calcium Nitrate*. Master thesis, Norwegian University of Science and Technology, Institute of Inorganic Chemistry, Trondheim, Norway, 1999. (in Norwegian).

[96] T. C. Powers. *The Properties of Fresh Concrete*. John Wiley & Sons, Inc., USA, 1968.

[97] T. C. Powers and E. M. Wiler. A device for studying the workability of concrete. In *Proc. ASTM*, volume 41, pages 1003–1015, 1941.

[98] V. S. Ramachandran. *Concrete Admixtures Handbook; Properties, Science, and Technology*. Noyes Publications, USA, 2nd edition, 1995.

[99] G. Ranalli. *Rheology of the Earth*. Chapman & Hall, Great Britain, 2nd edition, 1995.

[100] M. Reiner. *Deformation and Flow; An Elementary Introduction to Theoretical Rheology*. H. K. Lewis & Co. Limited, Great Britain, 1949.

[101] K. Reknes. Interaction between cement and lignosulfonate. In *Borregaard Symposium on Workability and Workability Retention*, Sarpsborg, Norway, 1998.

[102] K. Reknes. Reduction of viscosity of concrete with modified lignosulfonate. In Å. Skarendahl and Ö. Petersson, editors, *Self-Compacting Concrete, Proceedings of the First International RILEM Symposium*, Cachan Cedex, France, 1999. RILEM Publications S.A.R.L.

[103] K. Reknes. Set retardation and dispersing power of lignosulfonate and other materials. In *Borregaard Symposium on Retardation*, Løkken, Norway, 2000.

[104] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial Value Problems*. John Wiley, Inc., New York, USA, 1967.

[105] R. Rixom and N. Mailvaganam. *Chemical Admixture for Concrete*. F & FN Spon, New York, USA, 3rd edition, 1999.

[106] J. A. Roberson and C. T. Crowe. *Engineering Fluid Mechanics*. Houghton Mifflin Company, Boston, USA, 1990.

[107] T. Rojahn. Wood - source of lignin. In *Borregaard Symposium on Workability and Workability Retention*, Sarpsborg, Norway, 1998.

[108] W. G. V. Rosser. *An Introduction to Statistical Physics*. Ellis Horwood Limited, Great Britain, 1982.

[109] D. M. Roy. Portland cement: Constitution and processing; part i: Cement manufacture. In D. M. Roy, editor, *Instructional Modules in Cement Science*. Pennsylvania State University, Pennsylvania, USA, 1985.

[110] D. M. Roy. Portland cement: Constitution and processing; part ii: Cement constitution and kiln reactions. In D. M. Roy, editor, *Instructional Modules in Cement Science*. Pennsylvania State University, Pennsylvania, USA, 1985.

[111] K. L. Scrivener. *The Development of Microstructure During the Hydration of Portland Cement*. PhD thesis, University of London, London, Great Britain, 1984.

[112] K. L. Scrivener and P. L. Pratt. In *6th Proceedings of the International Conference of Cement Microscopy. International Cement Microscopy Association*, Duncanville, 1984.

[113] M. Shahinpoor. New constitutive equations for the rapid flow of granular materials - ii. *Journal of Non-Newtonian Fluid Mechanics*, 12:31–38, 1983.

[114] M. Shahinpoor and J. S. S. Siah. New constitutive equations for the rapid flow of granular materials. *Journal of Non-Newtonian Fluid Mechanics*, 9:147–156, 1981.

[115] D. J. Shaw. *Introduction to Colloid & Surface Chemistry*. Reed Educational and Professional Publishing Limited, Great Britain, 4^{th} edition, 1992.

[116] S. Smeplass. *High Strength Concrete; SP4 - Materials Design; Report 4.3: Heat of Hydration - A Parameter Study*. SINTEF, Cement and Concrete Research Institute, Trondheim, Norway, 1992. Report: STF70 A92131.

[117] W. A. Strauss. *Partial Differential Equations, An Introduction*. John Wiley & Sons, Inc., Singapore, 1992.

[118] M. S. Sumner, N. M. Herpher, and G. K. Moir. In *8th International Congress on the Chemistry of Cement*, Rio de Janeiro, 1986.

[119] U.S. Geological Survey. URL *http://www.usgs.gov*.

[120] Y. Tanigawa and H. Mori. Analytical study on deformation of fresh concrete. *Journal of Engineering Mechanics*, 115(3):493–508, 1989.

[121] Y. Tanigawa, H. Mori, K. Tsutsui, and Y. Kurokawa. Constitutive law and yield condition of fresh concrete. *Transactions of the Japan Concrete Institute*, 9:47–50, 1987.

[122] Y. Tanigawa, H. Mori, and K. Watanabe. Computer simulation of consistency and rheology tests of fresh concrete by viscoplastic finite element method. In *Proceedings of the RILEM Colloquium Properties of Fresh Concrete*, Hanover, 1990.

[123] R. I. Tanner and K. Walters. *Rheology: An Historical Perspective*. Elsevier Science B. V., Netherlands, 1998.

[124] E. J. Tarbuck and F. K. Lutgens. *Earth, An Introduction to Physical Geology*. Prentice-Hall, Inc., USA, 6^{th} edition, 1999.

[125] G. H. Tattersall. Relationships between the British standard test for workability and two-point test. *Magazine of Concrete Research*, 25(84):169–174, 1973.

[126] G. H. Tattersall. Progress in measurement of workability by the two-point test. In *Properties of Fresh Concrete, Proceedings of the RILEM Colloquium*, Great Britain, 1990. University Press, Cambridge.

[127] G. H. Tattersall. *Workability and Quality Control of Concrete*. E & FN Spon, London, Great Britain, 1991.

[128] G. H. Tattersall and P. F. G. Banfill. *The Rheology of Fresh Concrete*. Pitman Books Limited, Great Britain, 1983.

[129] G. H. Tattersall and Bloomer S. J. Further development of the two-point test for workability and extension of its range. *Magazine of Concrete Research*, 31(109):202–210, 1979.

[130] A. J. Taylor and S. D. R. Wilson. Conduit flow of an incompressible, yield-stress fluid. *Journal of Rheology*, 41(1):93–101, 1997.

[131] H. F. W. Taylor. *Cement Chemistry*. Thomas Telford Publishing, Great Britain, 2nd edition, 1997.

[132] N. L. Thomas and D. D. Double. Calcium and silicon concentrations in solution during the early hydration of Portland cement and tricalcium silicate. *Cement and Concrete Research*, 11:675–687, 1981.

[133] J. Topping. *Errors of Observation and Their Treatment*. Science Paperbacks, Great Britain, 4th edition, 1972.

[134] R. Trettin. Reactivity and mechanism of hydration of cement phases. In *Proceedings of the 10th International Congress on the Chemistry of Cement (10th ICCC), 2-6 June*, volume 2, Gothenburg, Sweden, 1997.

[135] H. Uchikawa. Hydration of cement and structure formation and properties of cement paste in the presence of organic admixture. In *Conference in Tribute to Micheline Moranville Regourd: "Importance of Recent Microstructural Developments in Cement and Concrete"*, Sherbrook University, Canada, 1994.

[136] H. Uchikawa, S. Hanehara, and D. Sawaki. A reply to the discussion by Susanta Chatterji of the paper "the role of steric repulsive force in the dispersion of cement particles in fresh paste prepared with organic admixture". *Cement and Concrete Research*, 27(9):1453–1455, 1997.

[137] H. Uchikawa, S. Hanehara, T. Shirasaka, and D. Sawaki. Effect of admixture on hydration of cement, adsorptive behavior of admixture and fluidity and setting of fresh cement paste. *Cement and Concrete Research*, 22(6):1115–1129, 1992.

[138] H. Uchikawa, D. Sawaki, and S. Hanehara. Influence of kind and added timing of organic admixture on the composition, structure and property of fresh cement paste. *Cement and Concrete Research*, 25(2):353–364, 1995.

[139] M. Ungarish. *Hydrodynamics of Suspensions*. Springer-Verlag, New York, USA, 1993.

[140] O. J. Uzomaka. A concrete rheometer and its application to a rheological study of concrete mixes. *Rheological Acta*, 13:12–21, 1974.

[141] E. J. W. Verwey and J. Th. G. Overbeek. *Theory of the Stability of Lyophobic Colloids*. Elsevier Publishing Co., Ltd., Amsterdam, Netherlands, 1948.

[142] J. E. Wallevik. *Implementation of the Sea Ice Model MCRIM 2.5 to Icelandic Waters*. The Icelandic Meteorological Office, Iceland, 1999. Report: VI-G99017-UR09 (in Icelandic).

[143] J. E. Wallevik. Rheological measurements on a high yield value bingham fluid with low agitation. In A. Friis and S. L. Mason, editors, *Annual Transactions of the Nordic Rheology Society*, volume 9, Trondheim, Norway, 2001.

[144] O. H. Wallevik. *Practical Rheological Description of the Fresh Concrete*. The Norwegian University of Science and Technology (Previously: The Norwegian Institute of Technology), Trondheim, Norway, 1989. Report: BML 89.201 (in Norwegian).

[145] O. H. Wallevik. *The Rheology of Fresh Concrete and its Application on Concrete with and without Silica Fume*. PhD thesis, The Norwegian University of Science and Technology (Previously: The Norwegian Institute of Technology), Trondheim, Norway, 1990. Dr.ing. thesis no. 1990:45 (in Norwegian).

[146] O. H. Wallevik. IBRI at Your Service; The Rheology of Fresh Concrete. Keldnaholti, IS-112 Reykjavik, Iceland, 1999.

[147] O. H. Wallevik and O. E. Gjørv. Development of a coaxial cylinder viscometer for fresh concrete. *Proceedings of the Rilem Colloquium: Properties of Fresh Concrete*, pages 213–224, 1990.

[148] R. E. Walpole, R. H. Myers, and S. L. Myers. *Probability and Statistics for Engineers and Scientists*. Prentice-Hall, Inc., New Jersey, USA, 6$^{\text{th}}$ edition, 1998.

[149] K. Wesche, H. Flatten, and W. vom Berg. The influence of separations on the rheological investigations of cement paste and mortar in the rotation rheometer. In *Proceedings of RILEM Seminar*, volume 1, 1973.

[150] M. Yang, C. M. Neubauer, and H. M. Jennings. Interparticle potential and sedimentation behavior of cement suspensions. *Advanced Cement Based Materials*, 5:1–7, 1997.

[151] J. F. Young. Influence of tricalcium aluminate on the hydration of calcium silicates. *Journal of the American Ceramic Society*, 52(1):44–46, 1969.

[152] J. F. Young. Hydration of portland cement. In D. M. Roy, editor, *Instructional Modules in Cement Science*. Pennsylvania State University, Pennsylvania, USA, 1985.

[153] M. Zeilik, S. A. Gregory, and E. V. P. Smith. *Introductory Astronomy and Astrophysics*. Harcourt Brace Jovanovich College Publishers, Saunders College Publishing, USA, 1992.

[154] S. S. Zumdahl. *Chemical Principles*. Houghton Mifflin Company, Boston, USA, 3$^{\text{rd}}$ edition, 1998.

Appendix A

Source Code

A.1 License Terms and Warranty

Two softwares are presented in this appendix. The first one is used when calculating the viscoplastic flow inside both ConTec viscometers (Appendix A.2). The second software is used when calculating the flow inside the C^3P^2-geometry (Appendix A.3). Both softwares will have the same name, namely Viscometric-ViscoPlastic-Flow (or VVPF). The reason for this is that both of them use much of the same or similar subroutines, stored as modules. The source codes are written in accordance with the **Fortran 90** standard (ANSI X3.198-1992 ∧ ISO/IEC 1539-1:1991 (E)) and as such, can be used on any computer platform having a Fortran 90 compiler. All the source files shown here are protected by the GNU General Public License (GNU GPL) as shown in the individual source code.

Viscometric-ViscoPlastic-Flow, is free software; which can be redistributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at the users option) any later version. Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License in Appendix A.4 for more details.

A.2 ConTec Viscometers

When calculating the viscoplastic flow inside the ConTec viscometers, Viscometric-ViscoPlastic-Flow consists of seven files listed below.

1. **param.f90** (MODULE): This code defines and sets all variables of relevance, like R_i, R_o, h, Δr, Δz, Δt, tol, tol_RMS, f_min, f_max and so forth.

2. **motion.f90** (MODULE): This file reads the basic information from **param.f90** to produce angular velocity $\omega_\mathrm{o} = \omega_\mathrm{o}(t)$. The information about the angular velocity is requested by the routine **main.f90**.

3. **viscous.f90** (MODULE): In this file, the shear viscosity function $\eta = \eta(\dot{\gamma}, t, \dots)$ is defined and calculated. This information is requested by **update.f90**.

4. **write2f.f90** (MODULE): This file takes care of writing all computed data into the different files. It is only the source **main.f90** that makes such request.

5. **shear.f90** (MODULE): This routine calculates the shear rate $\dot{\gamma}_c$ from the computed velocity profile V_k(i, j) $\mathbf{i}_\theta$. It is the program **update.f90** that makes the request.

6. **update.f90** (MODULE): This file sets up the system of algebraic Equations 7.28 to 7.31 (Page 161). This file also contains the Thomas algorithm that is used in solving this system.

7. **main.f90** (PROGRAM): This is the center of the whole software, holding and passing information to and from the different subroutines. Some subroutines interact directly with each other without going through the channels defined by **main.f90** (this applies mostly for the subroutines in the files **update.f90**, **shear.f90** and **viscous.f90**). The geometry of the viscometer, including the bottom cone, is defined in this part of the software.

The user only interacts with two files in the above list, namely the **param.f90** and **viscous.f90**. Hence, these files are explained in a more detail relative to other source codes. All the files can be compiled and linked in the order shown in the above.

A.2.1 ConTec BML Viscometer 3

Figure A.1 demonstrates the solution geometry and boundary conditions used when simulating the ConTec **BML** Viscometer 3. This array is 146×306 in size and the spacing of grid points in r and z-direction are uniform and equal: $\Delta r = \Delta z = 1.0\,\mathrm{mm}$. This greatly simplifies the programming of the software, saves storage space and usually results in greater accuracy in the numerical calculation [2]. In the algorithm, the viscometer is rotated by 90° clockwise to simplify array addressing. This is apparent when comparing Figure A.1 with the left illustration of Figure 8.15 (Page 197).

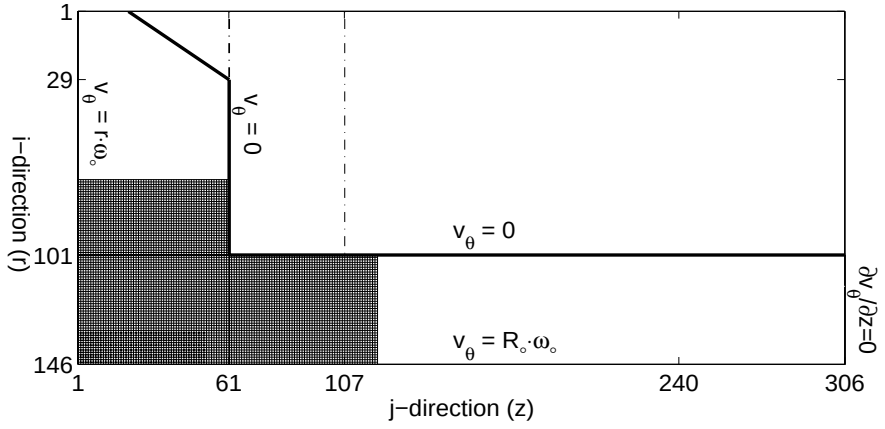


Figure A.1: Geometry and boundary conditions for the ConTec **BML** Viscometer 3. This information is put into the arrays (146 × 306) VELOCITY_k, VELOCITY_kp12 and VELOCITY_kp1 of the source code **main.f90**. The lower left corner demonstrates the grid resolution used. For the domain of calculation, there are about 19500 grid points.

Referring to Figure A.1, then in **i-direction** (i.e. in r-direction) the following numbers specify the geometry of the bottom cone in the viscometer: NX1_cone=1, NX_cone $\in [2, 26]$, NX2_cone=27. The parameters NX1_cone+1 and NX2_cone+1 designates the beginning and the end coordinates of the bottom cone. Also, the terms NX1 and NX2 designates the coordinates of the inner and outer cylinder and are given by 101 and 146, respectively. The geometry of the viscometer can be read directly from the figure. For example, the inner radius is $R_\mathrm{i} = (\mathrm{NX1} - 1)\Delta r = (101 - 1)1.0\,\mathrm{mm} = 10.0\,\mathrm{cm}$ and likewise the outer radius is $R_\mathrm{o} = (\mathrm{NX2} - 1)\Delta r = (146 - 1)1.0\,\mathrm{mm} = 14.5\,\mathrm{cm}$. All these variables are used in **main.f90** and **param.f90**.

In the **j-direction** (i.e. in z-direction) the corresponding *bottom-cone-numbers* are: NY1_cone=21, NY2_cone $\in [22, 60]$, where NY1_cone and last element of NY2_cone designates the beginning and the end coordinates of the bottom cone, respectively. The term NY1 designates the start location of the inner cylinder and NY2 designates the total height of the outer cylinder. These values are given by NY1=61 and NY2=306. The total height of the inner cylinder can now be calculated: $\mathrm{H2} = (\mathrm{NY2} - \mathrm{NY1})\Delta z = (306 - 61)1.0\,\mathrm{mm} = 24.5\,\mathrm{cm}$. The term $h \equiv \mathrm{H3} = 19.9\,\mathrm{cm}$ designates the height where torque is measured and it starts at the **j-grid** point $\mathrm{NY2mH3} = \mathrm{NY2} - \mathrm{H3}/\Delta z = 306 - 19.9\,\mathrm{cm}/1\,\mathrm{mm} = 107$ and ends at NY2=306.

A.2.2 ConTec Viscometer 4

Figure A.2 demonstrates the solution geometry and boundary conditions used when simulating the ConTec Viscometer 4. For the particular case shown, the solution array is 203×301 in size ($h \equiv \mathrm{H3} = 12.6\,\mathrm{cm}$). In Figure 8.2 (Page 180) it was chosen to be 203×281 ($h \equiv \mathrm{H3} = 11.6\,\mathrm{cm}$). In either case, the spacing of grid points in r and z-direction are uniform and equal: $\Delta r = \Delta z = 0.5\,\mathrm{mm}$.

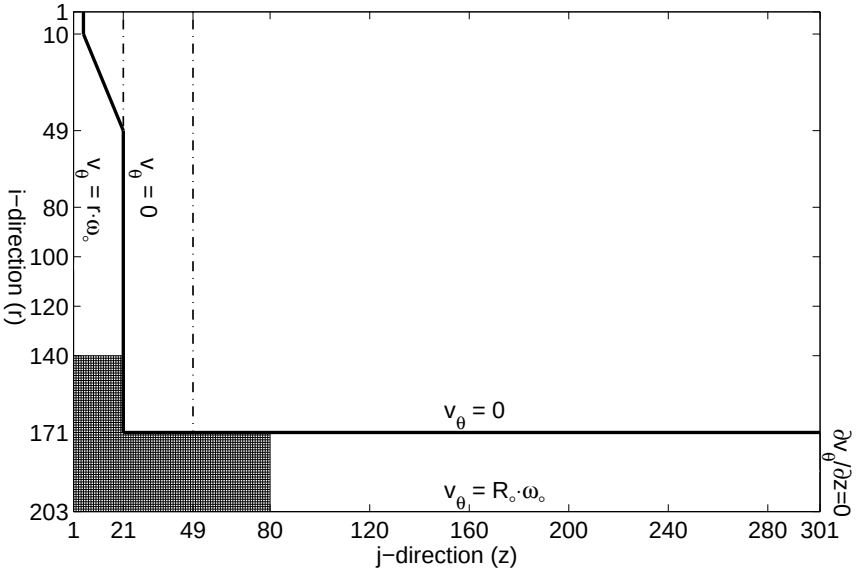


Figure A.2: Solution geometry and boundary conditions for the viscometer. This information is put into the arrays (203×301) VELOCITY_k, VELOCITY_kp12 and VELOCITY_kp1 in the routine **main.f90**. The lower left corner demonstrates the grid resolution used. For the domain of calculation, there are about 12000 grid points.

Referring to Figure A.2, then in **i-direction** (i.e. in r-direction) the following numbers specify the geometry of the bottom cone in the viscometer: NX1_cone=9, NX_cone $\in [12, 44]$, NX2_cone=47. The parameters NX1_cone+1 and NX2_cone+1 designates the beginning and the end coordinates of the bottom cone. Also, the terms NX1 and NX2 designates the coordinates of the inner and outer cylinder and are given by 171 and 203, respectively. Hence, the inner radius is $R_\mathrm{i} = (\mathrm{NX1} - 1)\Delta r = (171 - 1)0.5\,\mathrm{mm} = 8.5\,\mathrm{cm}$ and the outer radius is $R_\mathrm{o} = (\mathrm{NX2} - 1)\Delta r = (203 - 1)0.5\,\mathrm{mm} = 10.1\,\mathrm{cm}$.

In **j-direction** (i.e. in z-direction) the corresponding *bottom-cone-numbers* are: NY1_cone=5, NY2_cone $\in [6, 20]$, where NY1_cone and the last element of NY2_cone designates the beginning and the end coordinates of the bottom cone, respectively. The term NY1 designates the start location of the inner cylinder and NY2 designates the total height of the outer cylinder. These values are given by NY1=21 and NY2=301. The total height of the inner cylinder can now be calculated: $\mathrm{H2} = (\mathrm{NY2} - \mathrm{NY1})\Delta z = (301 - 21)0.5\,\mathrm{mm} = 14.0\,\mathrm{cm}$. As before, the term $h \equiv \mathrm{H3} = 12.6\,\mathrm{cm}$ designates the height where torque is measured and it starts at the **j-grid** point $\mathrm{NY2mH3} = \mathrm{NY2} - \mathrm{H3}/\Delta z = 301 - 12.6\,\mathrm{cm}/0.5\,\mathrm{mm} = 301 - 252 = 49$ and ends at NY2=301. In Section 8.3.2, this height was equal to $h \equiv \mathrm{H3} = 11.6\,\mathrm{cm}$ and hence the solution array for that particular case consisted of 203×281 instead of 203×301. All the coordinates related to the bottom cones are defined in **main.f90**. All other variables shown here are defined in **param.f90**. This applies also when calculating the viscoplastic flow inside the ConTec **BML** Viscometer 3 (Appendix A.2.1).

A.2.3 param.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: param.f90 (MODULE)                                                  !
! This code defines and sets all variables of relevance, like R_i, R_o, h, dr, dz,  !
! dt, tol, tol_RMS, f_min, f_max and so forth.                                   !
! The specific values for the different parameters shown here, corresponds to the   !
! case of VHMW Na, at t = 72 min and t = 102 min, in Section 9.5 (and Section 9.4). !
! ------------------------------------------------------------------------------ !
! rho => density of the test material in kg/m^3
! ------------
! ZERO_TIME (Section 7.11): maximum iteration time in seconds when solving the
! elliptical problem: Pseudotransient method is used. It will either be this
! time or the term "tol_Plastic" that will terminate the elliptical iteration.
! ------------
! "REAL_TIME": real time duration of iteration when solving the parabolic problem:
! ------------
```

```
! CALCULATE_TIME_DEPENDENT_PROBL: Inform the main routine if parabolic problem
! should be solved.
! ------------
! tol_Newton, tol_Plastic and tol_RMS: See Equation 7.73.
! tol => Tolerance to tell when to quit the successive substitution iteration and
! when, in the end, to quit iteration for the complete elliptical problem:
! Pseudotransient Method is used.
! ------------
! dt_Newton, dt_Plastic, dt => Time step. When solving for viscoplastic fluid,
! then much smaller time step is required compared to when solving for
! Newtonian fluid.
! ------------
! k_max = IDNINT(ZERO_TIME/dt_Plastic): Maximum amount of time steps
! when solving the elliptical problem.
! ------------
! NUMBER_OF_TIME_ITERATIONS = IDNINT(REAL_TIME/dt_Plastic)
! Number of time steps when solving the parabolic problem.
! ------------
! count_max => Maximum number of iterations for each successive substitution.
! ------------------------------------------------------------------------------ !
MODULE CONSTANTS_AND_PARAMETERS
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: WHAT_TYPE_OF_VISCOMETER,ConTec_CONSTANTS,VELOCITY_AND_TIME_ConTec,&
            BML_CONSTANTS,VELOCITY_AND_TIME_BML
CONTAINS
! ============================================================================== !
SUBROUTINE WHAT_TYPE_OF_VISCOMETER(ConTec_v4,ConTec_BML_v3)
! ------------
LOGICAL,INTENT(OUT) :: ConTec_v4,ConTec_BML_v3
! ------------
ConTec_v4     = .TRUE.
ConTec_BML_v3 = .FALSE.
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WHAT_TYPE_OF_VISCOMETER
! ============================================================================== !
SUBROUTINE ConTec_CONSTANTS(rho,REAL_TIME,CALCULATE_TIME_DEPENDENT_PROBL,&
                            ZERO_TIME,tol_Newton,tol_Plastic,tol_RMS,&
                            dt_Plastic,dt_Newton,count_max,R_i,R_o,h1,H2,H3)

DOUBLE PRECISION,INTENT(OUT) :: rho,REAL_TIME,ZERO_TIME,tol_Newton,&
                                tol_Plastic,tol_RMS,&
                                dt_Plastic,dt_Newton,R_i,R_o,h1,H2,H3
LOGICAL,INTENT(OUT)          :: CALCULATE_TIME_DEPENDENT_PROBL
INTEGER,INTENT(OUT)          :: count_max

DOUBLE PRECISION             :: TIME_INTERVAL,f,f_min,f_max,PERC
INTEGER                      :: NUMBER_OF_POINTS
LOGICAL                      :: SMOOTH
! ------------------------------------------------------------------------------ !
! Only ZERO_TIME and REAL_TIME are used from the
! call statement below:
CALL VELOCITY_AND_TIME_ConTec(ZERO_TIME,REAL_TIME,&
     TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)
! ------------------------------------------------------------
rho = 2090D0 ! Density of the test material in kg/m^3.
! ------------------------------------------------------------
! CALCULATE_TIME_DEPENDENT_PROBL = .FALSE.
CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.
! ------------------------------------------------------------
tol_Newton  = 0.5D-4  ! Used as condition for time independence in the Newtonian case.
                      ! Also used as tolerance for the successive substitution
                      ! (in this case it acts as a dummy variable since always two
                      ! successive steps are made for the Newtonian case).
tol_Plastic = 0.5D-5  ! For the successive substitution tolerance (Equation 7.73).
tol_RMS     = 1.0D-15 ! Condition for time independence (Equation 7.75).
! ------------------------------------------------------------
dt_Newton   = 1.0D-3
dt_Plastic  = 0.1D-4
count_max   = 15 ! Maximum number of successive (substitution) iterations, for each
                 ! time step k.
R_i = 0.085D0    ! =>  8.5 cm  = Inner radius of viscometer.
R_o = 0.101D0    ! => 10.1 cm  = Outer radius of viscometer.
```

```
h1  = 0.002D0    ! =>  0.2 cm  = Distance between bottom plate of viscometer and the
                 !               lowest part of cone.
H2  = 0.130D0    ! => 13.0 cm  = Total height of inner cylinder.
H3  = 0.116D0    ! => 11.6 cm  = Height where torque is measured (from top and downward).
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE ConTec_CONSTANTS
! ============================================================================== !
SUBROUTINE VELOCITY_AND_TIME_ConTec(ZERO_TIME,REAL_TIME,&
           TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)

DOUBLE PRECISION,INTENT(OUT) :: ZERO_TIME,REAL_TIME,&
                                TIME_INTERVAL,f,f_min,f_max,PERC
INTEGER,INTENT(OUT)          :: NUMBER_OF_POINTS
LOGICAL,INTENT(OUT)          :: SMOOTH
! ------------------------------------------------------------------------------ !
! If the condition "(DABS(VELOCITY_kp1(i,j) - VELOCITY_k(i,j)) > tol)", in
! "main.f90", never gets fulfilled, then it will be ZERO_TIME
! that will terminate the pseudotransient iteration (i.e. elliptical iteration).
ZERO_TIME = 0.5D0
! ------------------------------------------------------------
! Time interval for each constant angular velocity (in seconds).
! The total time is then TIME_INTERVAL*(NUMBER_OF_POINTS + 1)
!(Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.)
TIME_INTERVAL = 5.0D0
! ------------------------------------------------------------
! Total number of measuring points (up and down).
! This number must be an odd number, beginning with 3: 3,5,7,9,...
!(Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.)
NUMBER_OF_POINTS = 9
! ------------------------------------------------------------
! Total time of measurements (simulation), in seconds:
!(Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.)
REAL_TIME = TIME_INTERVAL*DBLE(NUMBER_OF_POINTS+1)
! ------------------------------------------------------------
! Only used if CALCULATE_TIME_DEPENDENT_PROBL = .FALSE.
f      = 0.10D0 ! -> Rotational frequency of the outer cylinder in 1/s (or rps).
! ------------------------------------------------------------
! Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.
f_min  = 0.10D0 ! -> Minimum rotational frequency of the outer cylinder in 1/s (or rps).
f_max  = 0.65D0 ! -> Maximum rotational frequency of the outer cylinder in 1/s (or rps).
PERC   = 0.18D0 !
SMOOTH = .TRUE. ! SMOOTH = .FALSE.
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE VELOCITY_AND_TIME_ConTec
! ============================================================================== !
SUBROUTINE BML_CONSTANTS(rho,REAL_TIME,CALCULATE_TIME_DEPENDENT_PROBL,&
                         ZERO_TIME,tol_Newton,tol_Plastic,tol_RMS,&
                         dt_Plastic,dt_Newton,count_max,R_i,R_o,h1,H2,H3)

DOUBLE PRECISION,INTENT(OUT) :: rho,REAL_TIME,ZERO_TIME,tol_Newton,&
                                tol_Plastic,tol_RMS,&
                                dt_Plastic,dt_Newton,R_i,R_o,h1,H2,H3
LOGICAL,INTENT(OUT)          :: CALCULATE_TIME_DEPENDENT_PROBL
INTEGER,INTENT(OUT)          :: count_max

DOUBLE PRECISION             :: TIME_INTERVAL,f,f_min,f_max,PERC
INTEGER                      :: NUMBER_OF_POINTS
LOGICAL                      :: SMOOTH
! ------------------------------------------------------------------------------ !
! Only ZERO_TIME and REAL_TIME are used from the
! call statement below:
CALL VELOCITY_AND_TIME_BML(ZERO_TIME,REAL_TIME,&
     TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)
! ------------------------------------------------------------
rho = 2354D0 ! kg/m^3
! ------------------------------------------------------------
CALCULATE_TIME_DEPENDENT_PROBL = .FALSE.
! CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.
! ------------------------------------------------------------
tol_Newton  = 1.0D-3  ! Used as condition for time independence in the Newtonian case.
                      ! Also used as tolerance for the successive substitution
                      ! (in this case it acts as a dummy variable since always two
```

```
                    ! successive steps are made for the Newtonian case).
tol_Plastic = 1.0D-10 ! For the successive substitution tolerance (Equation 7.73).
tol_RMS     = 1.0D-30 ! Condition for time independence (Equation 7.75).
! ------------------------------------------------------
dt_Newton   = 1.0D-1
dt_Plastic  = 0.1D-6
count_max   = 15 ! Maximum number of successive (substitution) iterations, for each
                 ! time step k.
R_i = 0.100D0    ! => 10.0 cm  = Inner radius of viscometer.
R_o = 0.145D0    ! => 14.5 cm  = Outer radius of viscometer.
h1  = 0.020D0    ! =>  2.0 cm  = Distance between bottom plate of viscometer and the
                 !               lowest part of cone.
H2  = 0.245D0    ! => 24.5 cm  = Total height of inner cylinder
H3  = 0.199D0    ! => 19.9 cm  = Height where torque is measured (from top and downward).
! ------------------------------------------------------------------------------- !
RETURN
END SUBROUTINE BML_CONSTANTS
! ============================================================================== !
SUBROUTINE VELOCITY_AND_TIME_BML(ZERO_TIME,REAL_TIME,&
           TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)

DOUBLE PRECISION,INTENT(OUT) :: ZERO_TIME,REAL_TIME,&
                                TIME_INTERVAL,f,f_min,f_max,PERC
INTEGER,INTENT(OUT)          :: NUMBER_OF_POINTS
LOGICAL,INTENT(OUT)          :: SMOOTH
! ------------------------------------------------------------------------------- !
! If the condition "(DABS(VELOCITY_kp1(i,j) - VELOCITY_k(i,j)) > tol)", in
! "main.f90", never gets fulfilled, then it will be ZERO_TIME
! that will terminate the pseudotransient iteration (i.e. elliptical iteration).
ZERO_TIME = 1.0D0
! ------------------------------------------------------
! Time interval for each constant angular velocity (in seconds).
! The total time is then TIME_INTERVAL*(NUMBER_OF_POINTS + 1)
!(Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.)
TIME_INTERVAL = 5.0D0
! ------------------------------------------------------
! Total number of measuring points (up and down).
! This number must be an odd number, beginning with 3: 3,5,7,9,...
!(Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.)
NUMBER_OF_POINTS = 9
! ------------------------------------------------------
! Total time of measurements (simulation), in seconds:
!(Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.)
REAL_TIME = TIME_INTERVAL*DBLE(NUMBER_OF_POINTS+1)
! ------------------------------------------------------
! Only used if CALCULATE_TIME_DEPENDENT_PROBL = .FALSE.
! f    = 0.50D0 ! -> Rotational frequency of the outer cylinder in 1/s (or rps).
f      = 3.0D0/(2.0D0*DACOS(-1.0D0)) ! -> omega = 3 rad/s (PI = DACOS(-1.0D0))
! ------------------------------------------------------
! Only used if CALCULATE_TIME_DEPENDENT_PROBL = .TRUE.
f_min  = 0.10D0 ! -> Minimum rotational frequency of the outer cylinder in 1/s (or rps).
f_max  = 0.65D0 ! -> Maximum rotational frequency of the outer cylinder in 1/s (or rps).
PERC   = 0.18D0 ! 0.30D0
SMOOTH = .TRUE. ! SMOOTH= .FALSE.
! ------------------------------------------------------------------------------- !
RETURN
END SUBROUTINE VELOCITY_AND_TIME_BML
! ============================================================================== !
END MODULE CONSTANTS_AND_PARAMETERS
! ------------------------------------------------------------------------------- !
```

A.2.4 viscous.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: viscous.f90 (MODULE)                                                !
! In this file, the shear viscosity function ETA = ETA(SR,t,...) is defined and  !
! calculated. This information is requested by update.f90.                       !
! ------------------------------------------------------------------------------ !
MODULE SHEAR_VISCOSITY
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: ETA,VISCOSITY,FMSR_FMCR
CONTAINS
! ============================================================================== !
SUBROUTINE ETA(dt,time,Lambda,SR_ij,SR_ip12j,SR_im12j,SR_ijp12,SR_ijm12,&
               FMSR_ij,FMSR_ip12j,FMSR_im12j,FMSR_ijp12,FMSR_ijm12,&
               FMCR_ij,FMCR_ip12j,FMCR_im12j,FMCR_ijp12,FMCR_ijm12,&
               ETA_ij,ETA_ip12j,ETA_im12j,ETA_ijp12,ETA_ijm12)
! ------------------------------------------------------------------------------ !
DOUBLE PRECISION,INTENT(IN)    :: dt,time,Lambda,&
                                  SR_ij,SR_ip12j,&
                                  SR_im12j,SR_ijp12,SR_ijm12,&
                                  FMSR_ij,FMSR_ip12j,&
                                  FMSR_im12j,FMSR_ijp12,FMSR_ijm12,&
                                  FMCR_ij,FMCR_ip12j,&
                                  FMCR_im12j,FMCR_ijp12,FMCR_ijm12

DOUBLE PRECISION,INTENT(OUT)   :: ETA_ij,ETA_ip12j,ETA_im12j,ETA_ijp12,&
                                  ETA_ijm12
! ------------------------------------------------------------------------------ !
CALL VISCOSITY(dt,time,Lambda,SR_ij,FMSR_ij,FMCR_ij,ETA_ij)
CALL VISCOSITY(dt,time,Lambda,SR_ip12j,FMSR_ip12j,FMCR_ip12j,ETA_ip12j)
CALL VISCOSITY(dt,time,Lambda,SR_im12j,FMSR_im12j,FMCR_im12j,ETA_im12j)
CALL VISCOSITY(dt,time,Lambda,SR_ijp12,FMSR_ijp12,FMCR_ijp12,ETA_ijp12)
CALL VISCOSITY(dt,time,Lambda,SR_ijm12,FMSR_ijm12,FMCR_ijm12,ETA_ijm12)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE ETA
! ============================================================================== !
SUBROUTINE VISCOSITY(dt,time,Lambda_tmp,SR,FMSR,FMCR,ETA)

DOUBLE PRECISION,INTENT(IN)    :: dt,time,Lambda_tmp,SR,FMSR,FMCR
DOUBLE PRECISION,INTENT(OUT)   :: ETA
DOUBLE PRECISION               :: mu,tau,a_1,a_2,mu_tmp,tau_tmp,delta,Lambda,&
                                  B3_n3_23,U_o,U_3,memory_alpha,memory_beta,H,&
                                  Gamma,Theta,BETA_I,BETA_II,ALPHA_I,ALPHA_II
! ------------------------------------------------------------------------------ !
CALL FMSR_FMCR(time,SR,U_o,memory_alpha,memory_beta,H)
! ------------------------------------------------------------------------------ !
! "Lambda >= 0" (Lambda_tmp.GT.-0.5D0) means that time independent calculations is
! present, with a constant shear viscosity ETA = constant. In this case, "Lambda" is
! related to the Continuation Method (see Section 7.8).
! "Lambda = -1" (Lambda_tmp.LT.-0.5D0) means that time dependent calculations
```

```
! (basically thixotropic) have begun with time dependent shear viscosity ETA = ETA(r,z,t).
! ------------------------------------------------------------------------------------ !
IF (Lambda_tmp.GT.-0.5D0) THEN        ! -> Time independent calculations.
  Lambda  = Lambda_tmp                !    Lambda = continuation parameter.
  Gamma   = 0.0D0
  Theta   = 0.0D0
ELSE IF (Lambda_tmp.LT.-0.5D0) THEN   ! -> Time dependent calculations.
  Lambda  = 1.0D0                     !    Lambda = dummy variable.
  ! ----
  ALPHA_I  = DEXP(time/memory_alpha)
  ALPHA_II = DEXP(-time/memory_alpha)
  Gamma    = ALPHA_II*(FMSR + ALPHA_I*SR*dt) ! Equations 7.69 to 7.70 (Equation 9.3)
  ! ----
  BETA_I   = DEXP(time/memory_beta)
  BETA_II  = DEXP(-time/memory_beta)
  Theta    = BETA_II*(FMCR + BETA_I*H*dt)    ! Equations 7.71 to 7.72 (Equation 9.4)
  Theta    = 0.0D0  ! <- This condition is only used for the VHMW Na-case. With
                    ! "Theta = 0.0D0", the re-coagulation is set equal to zero, as is
                    ! done in Sections 9.4 and 9.5. In Sections 9.6 to 9.8, the
                    ! re-coagulation is non-zero and hence, Theta is calculated
                    ! according to Equation 9.4 as "BETA_II*(FMCR + BETA_I*H*dt)".
END IF
! ------------------------------------------------------------------------------------ !
delta    =  0.005D0 ! <- The regularization parameter (see Section 7.9).
! ----------------------------------------------------------
! Values from Table 9.1 are shown here (t = 72 min and t = 102 min):
mu       =  0.650D0
a_1      =  1.100D0
B3_n3_23 = 30.000D0
tau      =  0.000D0
a_2      =  0.800D0
! ----------------------------------------------------------
! Equation 9.5:
U_3 = (U_o*(Theta*Gamma + 1.0D0) + Theta)/((Theta + 1.0D0)*(Gamma + 1.0D0))
! ----------------------------------------------------------
mu_tmp  = mu  + a_1*(B3_n3_23*(U_3**(2.0D0/3.0D0)))   ! Equation 9.8
tau_tmp = tau + a_2*(B3_n3_23*(U_3**(2.0D0/3.0D0)))   ! Equation 9.9
ETA     = mu_tmp + (tau_tmp*Lambda)/(SR + delta)      ! Equation 9.7
! ------------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE VISCOSITY
! ==================================================================================== !
SUBROUTINE FMSR_FMCR(time,SR,U_o,memory_alpha,memory_beta,H)

DOUBLE PRECISION,INTENT(IN)    :: time,SR
DOUBLE PRECISION,INTENT(OUT)   :: U_o,memory_alpha,memory_beta,H
DOUBLE PRECISION               :: k_1,k_2,k_3
! ------------------------------------------------------------------------------------ !
! As Theta = 0.0D0, then memory_beta is not used (Sections 9.4 and 9.5, only).
U_o          =  1.0D0
memory_alpha = 30.0D0
memory_beta  = 18.0D0
! ----------------------------------------------------------
! As Theta = 0.0D0, then k_1, k_2 and k_3 are not used (Sections 9.4 and 9.5, only).
k_1 = 0.005D0
k_2 = 0.100D0
k_3 = 0.005D0
! ----------------------------------------------------------
! Equations 9.10 and 9.11:
IF (time.EQ.0.0D0) THEN
  H = k_1*(1 - U_o)/4.0D0
ELSE IF ((time.GT. 0.0D0).AND.(time.LT.25.0D0)) THEN
  H = k_1/(SR**2 + 1.0D0)
ELSE IF ((time.GE.25.0D0).AND.(time.LT.45.0D0)) THEN
  H = k_2/(SR**2 + 1.0D0)
ELSE IF ((time.GE.45.0D0).AND.(time.LE.50.0D0)) THEN
  H = k_3/(SR**2 + 1.0D0)
END IF
! ------------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE FMSR_FMCR
! ==================================================================================== !
END MODULE SHEAR_VISCOSITY
! ------------------------------------------------------------------------------------ !
```

A.2.5 main.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: main.f90 (PROGRAM)                                                  !
! This is the center of the whole software, holding and passing information to and !
! from the different subroutines. Some subroutines interact directly with each   !
! other without going through the channels defined by main.f90 (this applies mostly !
! for the subroutines in the files update.f90, shear.f90 and viscous.f90).       !
! The geometry of the viscometer, including the bottom cone, is defined in this  !
! part of the software.                                                          !
! ------------------------------------------------------------------------------ !
PROGRAM MAIN_ROUTINE

USE CONSTANTS_AND_PARAMETERS
USE SHEAR_VISCOSITY
USE ROTATION
USE MATRIX
USE WRITE_INFORMATION

IMPLICIT NONE

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: MX1,MX2,MY1,MY2
DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: K1,K2,L1,L2,DUMMY_2

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: VELOCITY_k,VELOCITY_kp12,VELOCITY_kp12_new,&
                                               VELOCITY_kp1_new,VELOCITY_kp1,SR,H,FMSR,FMCR

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: v1r_ijp1k,v1r_ijk,v1r_ijm1k,&
                                               v1r_ijp1kp12,v1r_ijkp12,v1r_ijm1kp12,&
                                               v1r_c_ijp1k,v1r_c_ijk,v1r_c_ijm1k,&
                                               v1r_c_ijp1kp12,v1r_c_ijkp12,v1r_c_ijm1kp12,&
                                               v2r_ijp1k,v2r_ijk,v2r_ijm1k,&
                                               v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: FMSR1r_ijp1,FMSR1r_ij,FMSR1r_ijm1,&
                                               FMSR1r_c_ijp1,FMSR1r_c_ij,FMSR1r_c_ijm1,&
                                               FMSR2r_ijp1,FMSR2r_ij,FMSR2r_ijm1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: FMCR1r_ijp1,FMCR1r_ij,FMCR1r_ijm1,&
                                               FMCR1r_c_ijp1,FMCR1r_c_ij,FMCR1r_c_ijm1,&
                                               FMCR2r_ijp1,FMCR2r_ij,FMCR2r_ijm1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: v1r_ijkp12_new,v2r_ijkp12_new

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: v1z_ip1jkp12,v1z_ijkp12,v1z_im1jkp12,&
                                               v1z_ip1jkp1,v1z_ijkp1,v1z_im1jkp1,&
                                               v1z_c_ip1jkp12,v1z_c_ijkp12,v1z_c_im1jkp12,&
                                               v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
                                               v2z_ip1jkp12,v2z_ijkp12,v2z_im1jkp12,&
                                               v2z_ip1jkp1,v2z_ijkp1,v2z_im1jkp1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: FMSR2z_ip1j,FMSR2z_ij,FMSR2z_im1j,&
```

```
                                          FMSR1z_ip1j,FMSR1z_ij,FMSR1z_im1j,&
                                          FMSR1z_c_ip1j,FMSR1z_c_ij,FMSR1z_c_im1j

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: FMCR2z_ip1j,FMCR2z_ij,FMCR2z_im1j,&
                                          FMCR1z_ip1j,FMCR1z_ij,FMCR1z_im1j,&
                                          FMCR1z_c_ip1j,FMCR1z_c_ij,FMCR1z_c_im1j

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: v1z_ijkp1_new,v2z_ijkp1_new
INTEGER,ALLOCATABLE,DIMENSION(:)            :: x_cone,y_cone,y_cone_dynamic


DOUBLE PRECISION :: dr,dz,dt,rho,omega,Lambda,R_i,r_i_cone,R_o,h1,H2,H3,tol,a,b,&
                    dt_Newton,dt_Plastic,tol_Newton,tol_Plastic,tol_RMS,&
                    tol_RMS_active,ZERO_TIME,REAL_TIME,H_cone,h1_static,dt_OUTPUT,&
                    dt_OUTPUT_torque,RMS,vel_norm,small_zero,EPS,ZERO_TIME_tmp,&
                    REAL_TIME_tmp,TIME_INTERVAL,f,f_min,f_max,PERC,time,&
                    memory_beta,memory_alpha,BETA_I,ALPHA_I,SR_tmp,H_tmp,U_o

INTEGER          :: i,j,k,error,problem,NX,NX1,NX2,NY1,NY2,N_Lambda,N_Lambda_MAX,&
                    count,MAX_NUMBER_OF_ITERATIONS,count_max,NX1_Cone,&
                    NX2_Cone,NX_cone,NY1_Cone,NY2_cone,NX2mNX_cone,&
                    NUMBER_OF_TIME_ITERATIONS,N_cone_x,N_cone_y,&
                    NUMBER_OF_POINTS,N_dt,count_rms,NX1_cone_rms,k_OUTPUT_rms,&
                    k_OUTPUT_torque,k_OUTPUT,NY2mH3,DUMMY_1

LOGICAL          :: CONVERGENCE,TIME_INDEPENDENCE,WARNING_SIGN,ConTec_v4,&
                    ConTec_BML_v3,CALCULATE_TIME_DEPENDENT_PROBL,SMOOTH,&
                    FALSE_CONVERGENCE

CHARACTER        :: IGNORED_INPUT
! ------------------------------------------------------------------------------- !
PRINT *, "      ------------------------------------------------      "
PRINT *, "      Viscometric-ViscoPlastic-Flow v1.0  (CT3 & CT4)       "
PRINT *, "                                                            "
PRINT *, "             Copyright (C) 2002, Jon E. Wallevik,           "
PRINT *, "                 (jon.wallevik@bygg.ntnu.no)                "
PRINT *, " The Norwegian University of Science and Technology (NTNU)  "
PRINT *, "             Department of Structural Engineering           "
PRINT *, " ------------------------------------------------------------ "
PRINT *, "                                                            "
PRINT *, " This software is free software; you can redistribute it    "
PRINT *, " and/or modify it under the terms of the GNU General Public "
PRINT *, " License as published by the Free Software Foundation;      "
PRINT *, " either version 2 of the License, or (at your option) any   "
PRINT *, " later version. This software is distributed in the hope    "
PRINT *, " that it will be useful, but WITHOUT ANY WARRANTY; without  "
PRINT *, " even the implied warranty of MERCHANTABILITY or FITNESS    "
PRINT *, " FOR A PARTICULAR PURPOSE.                                  "
PRINT *, " See the GNU General Public License for more details.       "
PRINT *, " ------------------------------------------------------------ "
PRINT *, "                                                            "
WRITE(*,"(A)",ADVANCE="NO") "               PRESS 'ENTER' TO CONTINUE"
PRINT *, " "
READ (*,"(A)") IGNORED_INPUT
PRINT *, " "
! ------------------------------------------------------------------------------- !
CALL WHAT_TYPE_OF_VISCOMETER(ConTec_v4,ConTec_BML_v3)
! -------------------------------------------------------
IF ((ConTec_v4).AND.(ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .TRUE.  "
  STOP
ELSE IF ((.NOT.ConTec_v4).AND.(.NOT.ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .FALSE. "
  STOP
END IF
! ------------------------------------------------------------------------------- !
IF (ConTec_v4) THEN
  PRINT '(7X,A43)', "=========================================="
  PRINT '(7X,A43)', "      Solving for ConTec Viscometer 4      "
  PRINT '(7X,A43)', "=========================================="
  PRINT '(7X,A43)', "                                           "

  CALL ConTec_CONSTANTS(rho,REAL_TIME,CALCULATE_TIME_DEPENDENT_PROBL,&
                        ZERO_TIME,tol_Newton,tol_Plastic,tol_RMS,&
```

```
                      dt_Plastic,dt_Newton,count_max,R_i,R_o,h1,H2,H3)

  CALL VELOCITY_AND_TIME_ConTec(ZERO_TIME_tmp,REAL_TIME_tmp,&
       TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)

  dr = 0.5D-3 ! => 0.5 mm = Spacing between grid points in r-direction.
  dz = 0.5D-3 ! => 0.5 mm = Spacing between grid points in z-direction.

  H_cone    = DBLE(21 - 5)*dz ! -> Static (non changeable) variable:: 0.8 cm.
  h1_static = DBLE(5 - 1)*dz  ! -> Static (non changeable) variable:: 0.2 cm.

  N_cone_x  = 16              ! -> Static (non changeable) variable
  N_cone_y  = 39              ! -> Static (non changeable) variable

  ALLOCATE(x_cone(N_cone_x),y_cone(N_cone_y),y_cone_dynamic(N_cone_y),&
           stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
    PRINT *," Error code 0 in main and execution terminated!           "
    STOP
  END IF

  ! NX1=171;NX2=203;NY1=21;NY2=301;NX=NX2-NX1+1=203-171+1=33;
  x_cone   = (/9,12,14,17,19,22,24,27,29,32,34,37,39,42,44,47/)
  NX1_cone = x_cone(1)
  NX2_cone = x_cone(16)

  y_cone = (/5,6,6,6,7,7,8,8,8,9,9,10,10,10,11,11,12,12,12,13,13,&
           14,14,14,15,15,16,16,16,17,17,18,18,18,19,19,20,20,20/)
  y_cone_dynamic = y_cone - IDNINT((h1_static - h1)/dz)
  y_cone         = y_cone_dynamic
  NY1_cone       = y_cone(1)
! NY2_cone       = is defined elsewhere!
! ------------------------------------------------------------------------------ !
ELSE IF (ConTec_BML_v3) THEN
  PRINT '(4X,A49)', "================================================="
  PRINT '(4X,A49)', "       Solving for ConTec BML Viscometer 3       "
  PRINT '(4X,A49)', "================================================="
  PRINT '(4X,A49)', "                                                 "

  CALL BML_CONSTANTS(rho,REAL_TIME,CALCULATE_TIME_DEPENDENT_PROBL,&
                     ZERO_TIME,tol_Newton,tol_Plastic,tol_RMS,&
                     dt_Plastic,dt_Newton,count_max,R_i,R_o,h1,H2,H3)

  CALL VELOCITY_AND_TIME_BML(ZERO_TIME_tmp,REAL_TIME_tmp,&
       TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)

  dr = 1.0D-3 ! => 1.0 mm = Spacing between grid points in r-direction.
  dz = 1.0D-3 ! => 1.0 mm = Spacing between grid points in z-direction.

  H_cone    = DBLE(61 - 21)*dz ! -> Static (non changeable) variable:: 4.0 cm.
  h1_static = DBLE(21 - 1)*dz  ! -> Static (non changeable) variable:: 2.0 cm.

  N_cone_x = 40
  N_cone_y = 27

  ALLOCATE(x_cone(N_cone_x),y_cone(N_cone_y),y_cone_dynamic(N_cone_y),&
           stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
    PRINT *," Error code 0 in main and execution terminated!           "
    STOP
  END IF

  ! NX1=101;NX2=146;NY1=61;NY2=306;NX=NX2-NX1+1=146-101+1=46;
  x_cone   = (/1,2,2,3,4,4,5,6,6,7,8,8,9,10,10,11,12,12,13,14,14,15,&
            16,16,17,18,18,19,20,20,21,22,22,23,24,24,25,26,26,27/)
  NX1_cone = x_cone(1)
  NX2_cone = x_cone(40)

  y_cone   = (/21,22,24,25,27,28,30,31,33,34,36,37,39,40,42,43,&
            45,46,48,49,51,52,54,55,57,58,60/)
  y_cone_dynamic = y_cone - IDNINT((h1_static - h1)/dz)
  y_cone         = y_cone_dynamic
```

```
  NY1_cone       = y_cone(1)
! NY2_cone       = is defined elsewhere!

END IF
! ------------------------------------------------------------------------------ !
NX1    = IDNINT(R_i/dr) + 1 ! 171 for CT and 101 for BML
NX2    = IDNINT(R_o/dr) + 1 ! 203 for CT and 146 for BML
NX     = NX2 - NX1 + 1      ! NX=203-171+1=33 for CT and NX=146-101+1=46 for BML
NY1    = IDNINT((h1 + H_cone)/dz) + 1      ! = 21  for CT and 61  for BML
NY2    = IDNINT((h1 + H_cone + H2)/dz) + 1 ! = 301 for CT and 306 for BML
NY2mH3 = NY2 - IDNINT(H3/dz)

R_i = dr*DBLE(NX1-1)
R_o = dr*DBLE(NX2-1)
! ------------------------------------------------------------------------------ !
CALL ANGULAR_VELOCITY(0.0D0,dt,omega)
! ------------------------------------------------------
14 FORMAT(10X,"NX1 = ",(I3,2X),"; NX2 = ",(I3,2X),"; dr = ",F6.4,"m")
15 FORMAT(10X,"R_i = ",F6.4,"m  ; R_o = ",F6.4,"m")
19 FORMAT(10X,"NY2mH3 = ",(I3,2X)," ; NX = ",(I3,2X))
16 FORMAT(10X,"NY1 = ",(I3,2X),"; NY2 = ",(I3,2X),"; dz = ",F6.4,"m")
17 FORMAT(10X,"dt_Plastic = ",E9.3,"s; f_o = ",F6.4,"rps")
PRINT '(8X,A26)',"Geometric and time values:"
PRINT 14, NX1,NX2,dr
PRINT 16, NY1,NY2,dz
PRINT 19, NY2mH3,NX
PRINT 15, R_i,R_o
PRINT 17, dt_Plastic,omega/(2.0D0*ACOS(-1.0D0))
PRINT *, "    "
! ------------------------------------------------------------------------------ !
IF (NX1.GE.NX2) THEN
  PRINT *, " Inner radius 'R_i' is larger than the outer radius 'R_o'! "
  PRINT *, "              TERMINAL ERROR!                               "
  STOP
END IF

IF ((NX2_cone+2).GE.NX1) THEN
  PRINT *, " Inner radius 'R_i' is smaller than the largest radius "
  PRINT *, " of the bottom cone r_cone^max=((NX2_cone+2)-1)*dr!    "
  PRINT *, "              TERMINAL ERROR!                          "
  STOP
END IF

IF (h1.LT.2.0D0*dz) THEN
  PRINT *, " The variable 'h1' in 'param.f90' is too small! "
  PRINT *, " Minimum height of h1 must be 2*dz,             "
  PRINT *, " otherwise a logical error will occur.          "
  PRINT *, "  [h1(ConTec) = 0.001 (0.1 cm) ; dz=0.5 mm];    "
  PRINT *, "  [h1(BML)    = 0.002 (0.2 cm) ; dz=1.0 mm];    "
  PRINT *, "              TERMINAL ERROR!                   "
  STOP
END IF

IF (H2.LT.2.0D0*dz) THEN
  PRINT *, " The variable 'H2' in 'param.f90' is too small! "
  PRINT *, " Minimum height of h1 must be 2*dz,             "
  PRINT *, " otherwise a logical error will occur.          "
  PRINT *, "  [h1(ConTec) = 0.001 (0.1 cm) ; dz=0.5 mm];    "
  PRINT *, "  [h1(BML)    = 0.002 (0.2 cm) ; dz=1.0 mm];    "
  PRINT *, "              TERMINAL ERROR!                   "
  STOP
END IF
! ------------------------------------------------------------------------------ !
CALL WARNING_FOR_WRITING(NY2)

MAX_NUMBER_OF_ITERATIONS  = IDNINT(ZERO_TIME/dt_Plastic)
NUMBER_OF_TIME_ITERATIONS = DINT(REAL_TIME/dt_Plastic)

PRINT *, "      ------------------------------------------  "
PRINT "( '         MAX_NUMBER_OF_ITERATIONS: ', I10 )        ",&
                   MAX_NUMBER_OF_ITERATIONS
PRINT *, "      ------------------------------------------  "
PRINT *, "                                                  "
```

```
IF (CALCULATE_TIME_DEPENDENT_PROBL) THEN
  PRINT *, "      ------------------------------------------  "
  PRINT "( '         NUMBER_OF_TIME_ITERATIONS:       ', I10 ) ",&
                     NUMBER_OF_TIME_ITERATIONS
  PRINT *, "      ------------------------------------------  "
  PRINT *, "                                                  "
END IF

PRINT *, "               ---------------------------                  "
WRITE(*,"(A)",ADVANCE="NO") "                PRESS 'ENTER' TO CONTINUE"
PRINT *, " "
READ (*,"(A)") IGNORED_INPUT
PRINT *, " "
! ------------------------------------------------------------------------------ !
k_OUTPUT_rms = 25     ! -> Information output every dt_OUTPUT_rms
                      !    times (to console and file).
! ------------------------------------------------------------------------------ !
! The term "small_zero" does usually not have to be changed.
small_zero = 0.1D-7  ! -> Used in relation to screen and file output.
EPS        = 1.0D-15 ! -> Used in relation to vel_norm.
! ------------------------------------------------------------------------------ !
DUMMY_1    = IDNINT(0.8D0*DBLE(NY2))
! ------------------------------------------------------------------------------ !
! Creating log file and making the first entry:
OPEN(unit=8,file="log.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: log.dat! "
  STOP
ELSE
  WRITE (unit=8,fmt=*) 0,0.0D0
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
ALLOCATE(MX1(NX2-2,NX2-2),MX2(NX-2,NX-2),MY1(NY1-2,NY1-2),MY2(NY2-2,NY2-2),&
         VELOCITY_k(NX2,NY2),VELOCITY_kp12(NX2,NY2),VELOCITY_kp12_new(NX2,NY2),&
         VELOCITY_kp1(NX2,NY2),VELOCITY_kp1_new(NX2,NY2),SR(NX2,NY2),H(NX2,NY2),&
         FMSR(NX2,NY2),FMCR(NX2,NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 1 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(K1(NX2-2),K2(NX-2),L1(NY1-2),L2(NY2-2),DUMMY_2(NX),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 2 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(v1r_ijp1k(NX2),v1r_ijk(NX2),v1r_ijm1k(NX2),v1r_ijp1kp12(NX2),&
         v1r_ijkp12(NX2),v1r_ijm1kp12(NX2),v1r_c_ijp1k(NX2),v1r_c_ijk(NX2),&
         v1r_c_ijm1k(NX2),v1r_c_ijp1kp12(NX2),v1r_c_ijkp12(NX2),&
         v1r_c_ijm1kp12(NX2),v2r_ijp1k(NX),v2r_ijk(NX),v2r_ijm1k(NX),&
         v2r_ijp1kp12(NX),v2r_ijkp12(NX),v2r_ijm1kp12(NX),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 3a in main and execution terminated!          "
  STOP
END IF

ALLOCATE(FMSR1r_ijp1(NX2),FMSR1r_ij(NX2),FMSR1r_ijm1(NX2),&
         FMSR1r_c_ijp1(NX2),FMSR1r_c_ij(NX2),FMSR1r_c_ijm1(NX2),&
         FMSR2r_ijp1(NX),FMSR2r_ij(NX),FMSR2r_ijm1(NX),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 3b in main and execution terminated!          "
  STOP
END IF

ALLOCATE(FMCR1r_ijp1(NX2),FMCR1r_ij(NX2),FMCR1r_ijm1(NX2),&
         FMCR1r_c_ijp1(NX2),FMCR1r_c_ij(NX2),FMCR1r_c_ijm1(NX2),&
```

```
        FMCR2r_ijp1(NX),FMCR2r_ij(NX),FMCR2r_ijm1(NX),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 3c in main and execution terminated!          "
  STOP
END IF

ALLOCATE(v1r_ijkp12_new(NX2-2),v2r_ijkp12_new(NX-2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 4 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(v1z_ip1jkp12(NY1),v1z_ijkp12(NY1),v1z_im1jkp12(NY1),v1z_ip1jkp1(NY1),&
         v1z_ijkp1(NY1),v1z_im1jkp1(NY1),v1z_c_ip1jkp12(NY1),v1z_c_ijkp12(NY1),&
         v1z_c_im1jkp12(NY1),v1z_c_ip1jkp1(NY1),v1z_c_ijkp1(NY1),&
         v1z_c_im1jkp1(NY1),v2z_ip1jkp12(NY2),v2z_ijkp12(NY2),v2z_im1jkp12(NY2),&
         v2z_ip1jkp1(NY2),v2z_ijkp1(NY2),v2z_im1jkp1(NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 5a in main and execution terminated!          "
  STOP
END IF

ALLOCATE(FMSR1z_ip1j(NY1),FMSR1z_ij(NY1),FMSR1z_im1j(NY1),&
         FMSR1z_c_ip1j(NY1),FMSR1z_c_ij(NY1),FMSR1z_c_im1j(NY1),&
         FMSR2z_ip1j(NY2),FMSR2z_ij(NY2),FMSR2z_im1j(NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 5b in main and execution terminated!          "
  STOP
END IF

ALLOCATE(FMCR1z_ip1j(NY1),FMCR1z_ij(NY1),FMCR1z_im1j(NY1),&
         FMCR1z_c_ip1j(NY1),FMCR1z_c_ij(NY1),FMCR1z_c_im1j(NY1),&
         FMCR2z_ip1j(NY2),FMCR2z_ij(NY2),FMCR2z_im1j(NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 5c in main and execution terminated!          "
  STOP
END IF

ALLOCATE(v1z_ijkp1_new(NY1-2),v2z_ijkp1_new(NY2-2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 5d in main and execution terminated!          "
  STOP
END IF
! ------------------------------------------------------------------------------ !
! Initialization:
! ============================================================================== !
SR                = 0.0D0
H                 = 0.0D0
FMSR              = 0.0D0
FMCR              = 0.0D0
! ============================================================================== !
MX1               = 0.0D0
MX2               = 0.0D0
K1                = 0.0D0
K2                = 0.0D0

VELOCITY_k        = 0.0D0
VELOCITY_kp12     = 0.0D0
VELOCITY_kp12_new = 0.0D0
! ------------------------------------------------------------------------------ !
v1r_ijkp12_new    = 0.0D0
v2r_ijkp12_new    = 0.0D0

! v1r is used in K1 and MX1 -> K1=K1(v1r) and MX1=MX1(v1r) to solve the system
! MX1*v1r_new=K1. "v1r" could be called "v1r_old" since it is the velocity
! from the previous iteration.
v1r_ijp1k         = 0.0D0  ! v1r -> K1 & MX1*v1r_new=K1
v1r_ijk           = 0.0D0
```

```
v1r_ijm1k         = 0.0D0
v1r_ijp1kp12      = 0.0D0
v1r_ijkp12        = 0.0D0
v1r_ijm1kp12      = 0.0D0
! ------------------------------------------------------------------------------ !
FMSR1r_ijp1       = 0.0D0
FMSR1r_ij         = 0.0D0
FMSR1r_ijm1       = 0.0D0

FMCR1r_ijp1       = 0.0D0
FMCR1r_ij         = 0.0D0
FMCR1r_ijm1       = 0.0D0
! ------------------------------------------------------------------------------ !
v1r_c_ijp1k       = 0.0D0  ! v1r_c -> K1(1:x) & MX1(1:x)*v1r_c_new(1:x)=K1(1:x)
v1r_c_ijk         = 0.0D0  ! ..._c -> ..._cone
v1r_c_ijm1k       = 0.0D0
v1r_c_ijp1kp12    = 0.0D0
v1r_c_ijkp12      = 0.0D0
v1r_c_ijm1kp12    = 0.0D0
! ------------------------------------------------------------------------------ !
FMSR1r_c_ijp1     = 0.0D0
FMSR1r_c_ij       = 0.0D0
FMSR1r_c_ijm1     = 0.0D0

FMCR1r_c_ijp1     = 0.0D0
FMCR1r_c_ij       = 0.0D0
FMCR1r_c_ijm1     = 0.0D0
! ------------------------------------------------------------------------------ !
v2r_ijp1k         = 0.0D0  ! v2r -> K2 & MX2*v2r_new=K2
v2r_ijk           = 0.0D0
v2r_ijm1k         = 0.0D0
v2r_ijp1kp12      = 0.0D0
v2r_ijkp12        = 0.0D0
v2r_ijm1kp12      = 0.0D0
! ------------------------------------------------------------------------------ !
FMSR2r_ijp1       = 0.0D0
FMSR2r_ij         = 0.0D0
FMSR2r_ijm1       = 0.0D0

FMCR2r_ijp1       = 0.0D0
FMCR2r_ij         = 0.0D0
FMCR2r_ijm1       = 0.0D0
! ============================================================================== !
MY1               = 0.0D0
MY2               = 0.0D0
L1                = 0.0D0
L2                = 0.0D0

VELOCITY_kp1      = 0.0D0
VELOCITY_kp1_new  = 0.0D0
! ------------------------------------------------------------------------------ !
v1z_ijkp1_new     = 0.0D0
v2z_ijkp1_new     = 0.0D0

! v1z is used in L1 and MY1 -> L1=L1(v1z) and MY1=MY1(v1z) to solve the system
! MY1*v1z_new=L1. "v1z" could be called "v1z_old" since it is the velocity
! from the previous iteration.
v1z_ip1jkp12      = 0.0D0  ! v1z -> L1 & MY1*v1z_new=L1
v1z_ijkp12        = 0.0D0
v1z_im1jkp12      = 0.0D0
v1z_ip1jkp1       = 0.0D0
v1z_ijkp1         = 0.0D0
v1z_im1jkp1       = 0.0D0
! ------------------------------------------------------------------------------ !
FMSR1z_ip1j       = 0.0D0
FMSR1z_ij         = 0.0D0
FMSR1z_im1j       = 0.0D0

FMCR1z_ip1j       = 0.0D0
FMCR1z_ij         = 0.0D0
FMCR1z_im1j       = 0.0D0
! ------------------------------------------------------------------------------ !
v1z_c_ip1jkp12    = 0.0D0  ! v1z_c -> L1(1:y) & MY1(1:y)*v1z_c_new(1:y)=L1(1:y)
v1z_c_ijkp12      = 0.0D0  ! ..._c -> ..._cone
```

```
v1z_c_im1jkp12    = 0.0D0
v1z_c_ip1jkp1     = 0.0D0
v1z_c_ijkp1       = 0.0D0
v1z_c_im1jkp1     = 0.0D0
! ------------------------------------------------------------------------------ !
FMSR1z_c_ip1j     = 0.0D0
FMSR1z_c_ij       = 0.0D0
FMSR1z_c_im1j     = 0.0D0

FMCR1z_c_ip1j     = 0.0D0
FMCR1z_c_ij       = 0.0D0
FMCR1z_c_im1j     = 0.0D0
! ------------------------------------------------------------------------------ !
v2z_ip1jkp12      = 0.0D0  ! v2z -> L2 & MY2*v2z_new=L2
v2z_ijkp12        = 0.0D0
v2z_im1jkp12      = 0.0D0
v2z_ip1jkp1       = 0.0D0
v2z_ijkp1         = 0.0D0
v2z_im1jkp1       = 0.0D0
! ------------------------------------------------------------------------------ !
FMSR2z_ip1j       = 0.0D0
FMSR2z_ij         = 0.0D0
FMSR2z_im1j       = 0.0D0

FMCR2z_ip1j       = 0.0D0
FMCR2z_ij         = 0.0D0
FMCR2z_im1j       = 0.0D0
! ============================================================================== !
WARNING_SIGN      = .FALSE.
FALSE_CONVERGENCE = .FALSE.
! ------------------------------------------------------------------------------ !
! Initialization of boundary condition at t = 0.0 sec:
CALL ANGULAR_VELOCITY(0.0D0,dt,omega)
! Dirichlet boundary condition:
DO i = 1,NX2
  VELOCITY_k(i,1) = omega*DBLE(i-1)*dr
END DO
VELOCITY_k(1:NX1,NY1)       = 0.0D0
VELOCITY_k(1,1:NY1)         = 0.0D0
VELOCITY_k(NX1,NY1:NY2)     = 0.0D0
VELOCITY_k(NX2,:)           = R_o*omega

! ############################################################################## !
! In Section 7.11.1 is a detailed description of the algorithm, which is used in    !
! the following.                                                                     !
! ------------------------------------------------------------------------------ !
! Linear approximation to speed up convergence:
DO i = 2,NX1_cone+1
  a = VELOCITY_k(i,1)
  b = VELOCITY_k(i,NY1_cone)
  DO j = 2,NY1_cone-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(j-1)/DBLE(NY1_cone-1)
  END DO
END DO

DO i = NX1_cone+2,NX2_cone+1
  a = VELOCITY_k(i,1)
  b = VELOCITY_k(i,y_cone(i-NX1_cone))
  DO j = 2,y_cone(i-NX1_cone)-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(j-1)/DBLE(y_cone(i-NX1_cone)-1)
  END DO
END DO

DO i = NX2_cone+2,NX1
  a = VELOCITY_k(i,1)
  b = VELOCITY_k(i,NY1)
  DO j = 2,NY1-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(j-1)/DBLE(NY1-1)
  END DO
END DO

DO j = 2,NY2-1
  a = VELOCITY_k(NX1,j)
  b = VELOCITY_k(NX2,j)
```

```
  DO i = NX1+1,NX2-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(i-NX1)/DBLE(NX2-NX1)
  END DO
END DO

! Neumann boundary condition:
VELOCITY_k(NX1+1:NX2-1,NY2) = (4.0D0 * &
     VELOCITY_k(NX1+1:NX2-1,NY2-1) - &
     VELOCITY_k(NX1+1:NX2-1,NY2-2))/3.0D0

! CHECK OUT IF VELOCITY_k IS OK:
! CALL WRITE2FILE_k(VELOCITY_k,NX2)
! STOP
! ============================================================================== !
! ============================ Begin of CONTINUATION ============================ !
! ============================================================================== !
N_Lambda_MAX = 1
CONTINUATION: DO N_Lambda = 0,N_Lambda_MAX
! Lambda => The Continuation Method (see Section 7.8).
Lambda = DBLE(N_Lambda)/DBLE(N_Lambda_MAX)
PRINT *,"------------------------------------------------------"
PRINT *,"CONTINUATION:",Lambda

IF (N_Lambda == 0) THEN
  dt  = dt_Newton
  tol = tol_Newton
  tol_RMS_active = tol_Newton  ! -> See Equation 7.75.
ELSE
  dt  = dt_Plastic
  tol = tol_Plastic
  tol_RMS_active = tol_RMS     ! -> See Equation 7.75.
END IF

TIME_INDEPENDENCE = .FALSE.
! Initializing time for each CONTINUATION step:
k = 0
! ============================================================================== !
! ============================ Begin of the time loop =========================== !
! ============================================================================== !
ZERO_TIME_LOOP: DO WHILE (.NOT.TIME_INDEPENDENCE)
CONVERGENCE = .FALSE.
IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT *,"------------------------------------------------------"
  PRINT *,"                                                      "
  PRINT *,"  PSEUDO-TRANSIENT time step: k+1 = ",k+1
  PRINT *,"------------------------------------------------------"
END IF
! ---------------------------------------------------------------------------- !
! VELOCITY_kp12     = VELOCITY_k
! VELOCITY_kp1      = VELOCITY_k
! VELOCITY_kp1_new  = VELOCITY_kp1
! The following routines are to update the Dirichlet and Neumann boundary
! conditions for the time step k+1/2 and k+1. Most of these routines are
! redundant since the boundary conditions are not changing with time. However
! it is a good practice to include them, if by some unfortunate accident
! some of the boundary values are overwritten.
! ---------------------------------------------------------------------------- !
! A guess for the time step k+1/2
VELOCITY_kp12 = VELOCITY_k

! Updating boundary condition at k+1/2:
CALL ANGULAR_VELOCITY(0.0D0,dt,omega)

! Dirichlet boundary condition:
DO i = 1,NX2
  VELOCITY_kp12(i,1) = omega*DBLE(i-1)*dr
END DO
VELOCITY_kp12(1:NX1,NY1)       = 0.0D0
VELOCITY_kp12(1,1:NY1)         = 0.0D0
VELOCITY_kp12(NX1,NY1:NY2)     = 0.0D0
VELOCITY_kp12(NX2,:)           = R_o*omega

! Neumann boundary condition:
VELOCITY_kp12(NX1+1:NX2-1,NY2) = (4.0D0 * &
```

```
     VELOCITY_kp12(NX1+1:NX2-1,NY2-1) - &
     VELOCITY_kp12(NX1+1:NX2-1,NY2-2))/3.0D0

! Also updating boundary condition for "..._new":
VELOCITY_kp12_new = VELOCITY_kp12
! ------------------------------------------------------------------------------ !
! A guess for the time step k+1
VELOCITY_kp1 = VELOCITY_k

! Updating boundary condition at k+1:
CALL ANGULAR_VELOCITY(0.0D0,dt,omega)

! Dirichlet boundary condition:
DO i = 1,NX2
  VELOCITY_kp1(i,1) = omega*DBLE(i-1)*dr
END DO
VELOCITY_kp1(1:NX1,NY1)       = 0.0D0
VELOCITY_kp1(1,1:NY1)         = 0.0D0
VELOCITY_kp1(NX1,NY1:NY2)     = 0.0D0
VELOCITY_kp1(NX2,:)           = R_o*omega

! Neumann boundary condition:
VELOCITY_kp1(NX1+1:NX2-1,NY2) = (4.0D0 * &
     VELOCITY_kp1(NX1+1:NX2-1,NY2-1) - &
     VELOCITY_kp1(NX1+1:NX2-1,NY2-2))/3.0D0

! Also updating boundary condition for "..._new":
VELOCITY_kp1_new = VELOCITY_kp1
! ------------------------------------------------------------------------------ !
count = 0
! ============================================================================== !
! ====================== BEGIN OF SUCCESSIVE SUBSTITUTION ======================= !
! ============================================================================== !
! The iteration loop here is because of the non-linearity of the governing
! Equations 7.22 and 7.23. To come around this problem, the successive substitution
! approach is used (see Section 7.8).
CONVERGE: DO WHILE (.NOT.CONVERGENCE)

! If convergence is a problem, then this might help:
! VELOCITY_kp12 = (VELOCITY_k + VELOCITY_kp1_new)/2
count = count + 1
10 FORMAT(4X,"Successive substitution number = ",1(I3,1X))

IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT 10, count
END IF

!======================== BEGIN OF X-ITERATION ========================
! Iteration is made along r-direction (i.e. along the i-direction as
! in A(i,j)). It starts at the bottom of the viscometer i=(2:NX2-1) at
! j = 2 and then move upward with increasing j (see Figures 8.1 and 8.2).

DO j = 2,NY1_cone-1
  v1r_ijp1k    = VELOCITY_k(:,j+1)
  v1r_ijk      = VELOCITY_k(:,j)
  v1r_ijm1k    = VELOCITY_k(:,j-1)
  v1r_ijp1kp12 = VELOCITY_kp12(:,j+1)
  v1r_ijkp12   = VELOCITY_kp12(:,j)
  v1r_ijm1kp12 = VELOCITY_kp12(:,j-1)

  FMSR1r_ijp1  = FMSR(:,j+1)
  FMSR1r_ij    = FMSR(:,j)
  FMSR1r_ijm1  = FMSR(:,j-1)

  FMCR1r_ijp1  = FMCR(:,j+1)
  FMCR1r_ij    = FMCR(:,j)
  FMCR1r_ijm1  = FMCR(:,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,0.0D0,dz,NX2,v1r_ijp1k,v1r_ijk,&
                       v1r_ijm1k,v1r_ijp1kp12,v1r_ijkp12,v1r_ijm1kp12,&
                       FMSR1r_ijp1,FMSR1r_ij,FMSR1r_ijm1,&
                       FMCR1r_ijp1,FMCR1r_ij,FMCR1r_ijm1,MX1,K1)

  CALL MATRIX_SOLVER(MX1,K1,v1r_ijkp12_new,NX2-2)
```

```
  VELOCITY_kp12_new(2:NX2-1,j) = v1r_ijkp12_new
END DO

! -------------------------- BEGIN OF X-CONE ------------------------
DO j = NY1_cone,NY1-1
  NX_cone                       = x_cone(j + 1 - NY1_cone)
  r_i_cone                      = NX_cone*dr
  NX2mNX_cone                   = NX2 - NX_cone
  v1r_c_ijp1k(1:NX2mNX_cone)    = VELOCITY_k(NX_cone+1:NX2,j+1)
  v1r_c_ijk(1:NX2mNX_cone)      = VELOCITY_k(NX_cone+1:NX2,j)
  v1r_c_ijm1k(1:NX2mNX_cone)    = VELOCITY_k(NX_cone+1:NX2,j-1)
  v1r_c_ijp1kp12(1:NX2mNX_cone) = VELOCITY_kp12(NX_cone+1:NX2,j+1)
  v1r_c_ijkp12(1:NX2mNX_cone)   = VELOCITY_kp12(NX_cone+1:NX2,j)
  v1r_c_ijm1kp12(1:NX2mNX_cone) = VELOCITY_kp12(NX_cone+1:NX2,j-1)

  FMSR1r_c_ijp1(1:NX2mNX_cone)  = FMSR(NX_cone+1:NX2,j+1)
  FMSR1r_c_ij(1:NX2mNX_cone)    = FMSR(NX_cone+1:NX2,j)
  FMSR1r_c_ijm1(1:NX2mNX_cone)  = FMSR(NX_cone+1:NX2,j-1)

  FMCR1r_c_ijp1(1:NX2mNX_cone)  = FMCR(NX_cone+1:NX2,j+1)
  FMCR1r_c_ij(1:NX2mNX_cone)    = FMCR(NX_cone+1:NX2,j)
  FMCR1r_c_ijm1(1:NX2mNX_cone)  = FMCR(NX_cone+1:NX2,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,r_i_cone,dz,NX2mNX_cone,v1r_c_ijp1k,v1r_c_ijk,&
                       v1r_c_ijm1k,v1r_c_ijp1kp12,v1r_c_ijkp12,v1r_c_ijm1kp12,&
                       FMSR1r_c_ijp1,FMSR1r_c_ij,FMSR1r_c_ijm1,&
                       FMCR1r_c_ijp1,FMCR1r_c_ij,FMCR1r_c_ijm1,MX1,K1)

  CALL MATRIX_SOLVER(MX1,K1,v1r_ijkp12_new,NX2mNX_cone-2)
! CALL WRITE2FILE_debug(MX1,K1,v1r_ijkp12_new,NX2mNX_cone-2)
! STOP
  VELOCITY_kp12_new(NX_cone+2:NX2-1,j) = v1r_ijkp12_new(1:NX2mNX_cone-2)
END DO
! --------------------------- END OF X-CONE -------------------------

DO j = NY1,NY2-1
  v2r_ijp1k    = VELOCITY_k(NX1:NX2,j+1)
  v2r_ijk      = VELOCITY_k(NX1:NX2,j)
  v2r_ijm1k    = VELOCITY_k(NX1:NX2,j-1)
  v2r_ijp1kp12 = VELOCITY_kp12(NX1:NX2,j+1)
  v2r_ijkp12   = VELOCITY_kp12(NX1:NX2,j)
  v2r_ijm1kp12 = VELOCITY_kp12(NX1:NX2,j-1)

  FMSR2r_ijp1  = FMSR(NX1:NX2,j+1)
  FMSR2r_ij    = FMSR(NX1:NX2,j)
  FMSR2r_ijm1  = FMSR(NX1:NX2,j-1)

  FMCR2r_ijp1  = FMCR(NX1:NX2,j+1)
  FMCR2r_ij    = FMCR(NX1:NX2,j)
  FMCR2r_ijm1  = FMCR(NX1:NX2,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,v2r_ijp1k,v2r_ijk,&
                       v2r_ijm1k,v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12,&
                       FMSR2r_ijp1,FMSR2r_ij,FMSR2r_ijm1,&
                       FMCR2r_ijp1,FMCR2r_ij,FMCR2r_ijm1,MX2,K2)

  CALL MATRIX_SOLVER(MX2,K2,v2r_ijkp12_new,NX-2)
  VELOCITY_kp12_new(NX1+1:NX2-1,j) = v2r_ijkp12_new
END DO

j = NY2
v2r_ijp1k    = VELOCITY_k(NX1:NX2,j-1)    ! => v(i,j+1) = v(i,j-1)
v2r_ijk      = VELOCITY_k(NX1:NX2,j)
v2r_ijm1k    = VELOCITY_k(NX1:NX2,j-1)
v2r_ijp1kp12 = VELOCITY_kp12(NX1:NX2,j-1) ! => v(i,j+1) = v(i,j-1)
v2r_ijkp12   = VELOCITY_kp12(NX1:NX2,j)
v2r_ijm1kp12 = VELOCITY_kp12(NX1:NX2,j-1)

FMSR2r_ijp1  = FMSR(NX1:NX2,j-1)          ! => v(i,j+1) = v(i,j-1)
FMSR2r_ij    = FMSR(NX1:NX2,j)
FMSR2r_ijm1  = FMSR(NX1:NX2,j-1)

FMCR2r_ijp1  = FMCR(NX1:NX2,j-1)          ! => v(i,j+1) = v(i,j-1)
FMCR2r_ij    = FMCR(NX1:NX2,j)
```

```
FMCR2r_ijm1  = FMCR(NX1:NX2,j-1)

CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,v2r_ijp1k,v2r_ijk,&
                     v2r_ijm1k,v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12,&
                     FMSR2r_ijp1,FMSR2r_ij,FMSR2r_ijm1,&
                     FMCR2r_ijp1,FMCR2r_ij,FMCR2r_ijm1,MX2,K2)

CALL MATRIX_SOLVER(MX2,K2,v2r_ijkp12_new,NX-2)
VELOCITY_kp12_new(NX1+1:NX2-1,j) = v2r_ijkp12_new

! --------- PAUSE FOR DEBUGGING ---------
! CALL WRITE2FILE_k(VELOCITY_kp12_new,NX2)
! WRITE(*,"(A)",ADVANCE="NO") " PRESS 'ENTER' TO CONTINUE "
! PRINT *, " "
! READ (*,"(A)") IGNORED_INPUT
! PRINT *, " "

!======================== END OF X-ITERATION ========================

! Updating ..._kp12:
VELOCITY_kp12 = VELOCITY_kp12_new

!======================== BEGIN OF Y-ITERATION ========================
! Iteration is made along z-direction (i.e. along the j-direction as
! in A(i,j)). It starts at the right side of the viscometer j=(2:NY2-1) at
! i = NX2-1 and then moves to the left with decreasing i
! (see Figures 8.1 and 8.2).
DO i = NX2-1,NX1+1,-1
  v2z_ip1jkp12 = VELOCITY_kp12(i+1,:)
  v2z_ijkp12   = VELOCITY_kp12(i,:)
  v2z_im1jkp12 = VELOCITY_kp12(i-1,:)
  v2z_ip1jkp1  = VELOCITY_kp1(i+1,:)
  v2z_ijkp1    = VELOCITY_kp1(i,:)
  v2z_im1jkp1  = VELOCITY_kp1(i-1,:)

  FMSR2z_ip1j  = FMSR(i+1,:)
  FMSR2z_ij    = FMSR(i,:)
  FMSR2z_im1j  = FMSR(i-1,:)

  FMCR2z_ip1j  = FMCR(i+1,:)
  FMCR2z_ij    = FMCR(i,:)
  FMCR2z_im1j  = FMCR(i-1,:)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY2,v2z_ip1jkp12,v2z_ijkp12,&
                       v2z_im1jkp12,v2z_ip1jkp1,v2z_ijkp1,v2z_im1jkp1,&
                       FMSR2z_ip1j,FMSR2z_ij,FMSR2z_im1j,&
                       FMCR2z_ip1j,FMCR2z_ij,FMCR2z_im1j,MY2,L2,.TRUE.)

  CALL MATRIX_SOLVER(MY2,L2,v2z_ijkp1_new,NY2-2)
  VELOCITY_kp1_new(i,2:NY2-1) = v2z_ijkp1_new
END DO

! v(i,j+1) = v(i,j-1) =>
VELOCITY_kp1_new(NX1+1:NX2-1,NY2) = VELOCITY_kp1_new(NX1+1:NX2-1,NY2-2)

DO i = NX1,NX2_cone+2,-1
  v1z_ip1jkp12 = VELOCITY_kp12(i+1,1:NY1)
  v1z_ijkp12   = VELOCITY_kp12(i,1:NY1)
  v1z_im1jkp12 = VELOCITY_kp12(i-1,1:NY1)
  v1z_ip1jkp1  = VELOCITY_kp1(i+1,1:NY1)
  v1z_ijkp1    = VELOCITY_kp1(i,1:NY1)
  v1z_im1jkp1  = VELOCITY_kp1(i-1,1:NY1)

  FMSR1z_ip1j  = FMSR(i+1,1:NY1)
  FMSR1z_ij    = FMSR(i,1:NY1)
  FMSR1z_im1j  = FMSR(i-1,1:NY1)

  FMCR1z_ip1j  = FMCR(i+1,1:NY1)
  FMCR1z_ij    = FMCR(i,1:NY1)
  FMCR1z_im1j  = FMCR(i-1,1:NY1)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY1,v1z_ip1jkp12,v1z_ijkp12,&
                       v1z_im1jkp12,v1z_ip1jkp1,v1z_ijkp1,v1z_im1jkp1,&
                       FMSR1z_ip1j,FMSR1z_ij,FMSR1z_im1j,&
```

```
                      FMCR1z_ip1j,FMCR1z_ij,FMCR1z_im1j,MY1,L1,.FALSE.)

  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY1-2)
  VELOCITY_kp1_new(i,2:NY1-1) = v1z_ijkp1_new
END DO

! -------------------------- BEGIN OF Y-CONE ------------------------
DO i = NX2_cone+1,NX1_cone+2,-1
  NY2_cone                    = y_cone(i-NX1_cone)
  v1z_c_ip1jkp12(1:NY2_cone)  = VELOCITY_kp12(i+1,1:NY2_cone)
  v1z_c_ijkp12(1:NY2_cone)    = VELOCITY_kp12(i,1:NY2_cone)
  v1z_c_im1jkp12(1:NY2_cone)  = VELOCITY_kp12(i-1,1:NY2_cone)
  v1z_c_ip1jkp1(1:NY2_cone)   = VELOCITY_kp1(i+1,1:NY2_cone)
  v1z_c_ijkp1(1:NY2_cone)     = VELOCITY_kp1(i,1:NY2_cone)
  v1z_c_im1jkp1(1:NY2_cone)   = VELOCITY_kp1(i-1,1:NY2_cone)

  FMSR1z_c_ip1j(1:NY2_cone)   = FMSR(i+1,1:NY2_cone)
  FMSR1z_c_ij(1:NY2_cone)     = FMSR(i,1:NY2_cone)
  FMSR1z_c_im1j(1:NY2_cone)   = FMSR(i-1,1:NY2_cone)

  FMCR1z_c_ip1j(1:NY2_cone)   = FMCR(i+1,1:NY2_cone)
  FMCR1z_c_ij(1:NY2_cone)     = FMCR(i,1:NY2_cone)
  FMCR1z_c_im1j(1:NY2_cone)   = FMCR(i-1,1:NY2_cone)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY2_cone,v1z_c_ip1jkp12,v1z_c_ijkp12,&
                       v1z_c_im1jkp12,v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
                       FMSR1z_c_ip1j,FMSR1z_c_ij,FMSR1z_c_im1j,&
                       FMCR1z_c_ip1j,FMCR1z_c_ij,FMCR1z_c_im1j,MY1,L1,.FALSE.)

  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY2_cone-2)
  VELOCITY_kp1_new(i,2:NY2_cone-1) = v1z_ijkp1_new(1:NY2_cone-2)
END DO

DO i = NX1_cone+1,2,-1
  v1z_c_ip1jkp12(1:NY1_cone)  = VELOCITY_kp12(i+1,1:NY1_cone)
  v1z_c_ijkp12(1:NY1_cone)    = VELOCITY_kp12(i,1:NY1_cone)
  v1z_c_im1jkp12(1:NY1_cone)  = VELOCITY_kp12(i-1,1:NY1_cone)
  v1z_c_ip1jkp1(1:NY1_cone)   = VELOCITY_kp1(i+1,1:NY1_cone)
  v1z_c_ijkp1(1:NY1_cone)     = VELOCITY_kp1(i,1:NY1_cone)
  v1z_c_im1jkp1(1:NY1_cone)   = VELOCITY_kp1(i-1,1:NY1_cone)

  FMSR1z_c_ip1j(1:NY1_cone)   = FMSR(i+1,1:NY1_cone)
  FMSR1z_c_ij(1:NY1_cone)     = FMSR(i,1:NY1_cone)
  FMSR1z_c_im1j(1:NY1_cone)   = FMSR(i-1,1:NY1_cone)

  FMCR1z_c_ip1j(1:NY1_cone)   = FMCR(i+1,1:NY1_cone)
  FMCR1z_c_ij(1:NY1_cone)     = FMCR(i,1:NY1_cone)
  FMCR1z_c_im1j(1:NY1_cone)   = FMCR(i-1,1:NY1_cone)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY1_cone,v1z_c_ip1jkp12,v1z_c_ijkp12,&
                       v1z_c_im1jkp12,v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
                       FMSR1z_c_ip1j,FMSR1z_c_ij,FMSR1z_c_im1j,&
                       FMCR1z_c_ip1j,FMCR1z_c_ij,FMCR1z_c_im1j,MY1,L1,.FALSE.)

  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY1_cone-2)
  VELOCITY_kp1_new(i,2:NY1_cone-1) = v1z_ijkp1_new(1:NY1_cone-2)
END DO
! -------------------------- END OF Y-CONE ------------------------

!======================== END OF Y-ITERATION ========================

CONVERGENCE = .TRUE.

! Settings for testing of convergence (or rather stability):
RMS      = 0.0D0
vel_norm = 1.0D0

S1: DO j = 2,NY1_cone-1
      DO i = 2,NX2_cone+2
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S1
```

```
        END IF
      END DO
    END DO S1

IF (CONVERGENCE) THEN
S2: DO j = NY1_cone,NY1-1
      NX1_cone_rms = x_cone(j-NY1_cone+1) + 2
      DO i = NX1_cone_rms,NX2_cone+2
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S2
        END IF
      END DO
    END DO S2
END IF

IF (CONVERGENCE) THEN
S3: DO j = 2,NY1-1
      DO i = NX2_cone+3,NX2-1
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S3
        END IF
      END DO
    END DO S3
END IF

IF (CONVERGENCE) THEN
S4: DO j = NY1,NY2
      DO i = NX1+1,NX2-1
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S4
        END IF
      END DO
    END DO S4
END IF

! Updating ..._kp1
VELOCITY_kp1 = VELOCITY_kp1_new

! CALL WRITE2FILE_k(VELOCITY_k,NX2)

IF (count == count_max) THEN
  CONVERGENCE       = .TRUE.
  FALSE_CONVERGENCE = .TRUE.
  PRINT *, " WARNING: FALSE CONVERGENCE! TIME STEP k = ", k
  PRINT *, " Maximum amount of successive substitutions is = ", count_max
  PRINT *, " ------------------------------------------------------------ "
  PRINT *, " RECOMMENDATION: Kill this application and reduce the time   "
  PRINT *, " step by an order of magnitude: dt -> dt/10                  "
  PRINT *, " ------------------------------------------------------------ "
END IF

END DO CONVERGE
! ============================================================================== !
! ====================== END OF SUCCESSIVE SUBSTITUTION ========================= !
! ============================================================================== !
! Checking for time independence:
count_rms = 0
RMS       = 0.0D0
vel_norm  = 1.0D0

DO j = 2,NY1_cone-1
  DO i = 2,NX2_cone+2
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
```

```
  END DO
END DO

DO j = NY1_cone,NY1-1
  NX1_cone_rms = x_cone(j-NY1_cone+1) + 2
  DO i = NX1_cone_rms,NX2_cone+2
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
  END DO
END DO

DO j = 2,NY1-1
  DO i = NX2_cone+3,NX2-1
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
  END DO
END DO

DO j = NY1,NY2
  DO i = NX1+1,NX2-1
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
  END DO
END DO

RMS = DSQRT(RMS/count_rms)

IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT *, "   RMS   =", RMS
  CALL WRITE2FILE_rms(k+1,RMS)
END IF

IF (RMS.LT.tol_RMS_active) THEN
  TIME_INDEPENDENCE = .TRUE.
ELSE
  TIME_INDEPENDENCE = .FALSE.
END IF

IF (k == MAX_NUMBER_OF_ITERATIONS) THEN
  TIME_INDEPENDENCE = .TRUE.
  WARNING_SIGN      = .TRUE.
END IF

VELOCITY_k = VELOCITY_kp1_new

k = k + 1

END DO ZERO_TIME_LOOP
! ============================================================================== !
! ============================ End of the time loop ============================ !
! ============================================================================== !
END DO CONTINUATION
! ============================================================================== !
! ============================== End of CONTINUATION ============================ !
! ============================================================================== !
IF (WARNING_SIGN) THEN
  PRINT *, " ------------------------------------------------------- "
  PRINT *, " WARNING: k = MAX_NUMBER_OF_ITERATIONS; See log.dat        "
  PRINT *, " RECOMMENDATIONS:                                         "
  PRINT *, " I)  Rerun this application with reduced time step.       "
  PRINT *, "     Try order of magnitude less: dt -> dt/10.            "
  PRINT *, " II) Increase ZERO_TIME in the file param.f90.            "
  PRINT *, " ------------------------------------------------------- "
END IF

IF (FALSE_CONVERGENCE) THEN
  PRINT *, " ------------------------------------------------------- "
  PRINT *, " WARNING: FALSE CONVERGENCE WAS ACHIEVED. Reduce the time  "
  PRINT *, " step by order of magnitude: dt -> dt/10 and then rerun    "
  PRINT *, " the application.                                         "
  PRINT *, " ------------------------------------------------------- "
```

```
END IF

PRINT *, " PSEUDO-TRANSIENT CALCULATION FINISHED! "
PRINT *, " ------------------------------------ "
! ------------------------------------------------------------------------------- !
IF (.NOT.CALCULATE_TIME_DEPENDENT_PROBL) THEN
  PRINT *, " Number of grid points (not including Dirichlet "
  PRINT *, " boundary points) = ", count_rms
  PRINT *, "                                               "
  PRINT *, " NO TIME DEPENDENT CALCULATION IS DONE SINCE    "
  PRINT *, " CALCULATE_TIME_DEPENDENT_PROBL = .FALSE.       "
  PRINT *, "                                               "
  PRINT *, " WRITING INFORMATION TO FILE, STAND BY...       "
  CALL WRITE2FILE_k(VELOCITY_k,NX2)
  CALL WRITE2FILE_kp1(VELOCITY_kp1_new,NX1,NY1,NX2,NY2,dr,dz)
  CALL WRITE2FILE_torque_ZERO(VELOCITY_k,NX1,NX2,NY1,NY2,0,dt,Lambda,dr,dz,H3,omega)
  PRINT *, " ...DONE! "

  ! CLEARING SOME MAJOR VARIABLES FROM THE RANDOM ACCESS MEMORY:
  DEALLOCATE(x_cone,y_cone,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 6 in main and execution terminated!             "
  END IF

  DEALLOCATE(MX1,MX2,MY1,MY2,VELOCITY_k,VELOCITY_kp12,VELOCITY_kp12_new,&
             VELOCITY_kp1,VELOCITY_kp1_new,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 7 in main!                                      "
  END IF

  DEALLOCATE(K1,K2,L1,L2,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 8 in main!                                      "
  END IF

  DEALLOCATE(v1r_ijp1k,v1r_ijk,v1r_ijm1k,v1r_ijp1kp12,v1r_ijkp12,&
             v1r_ijm1kp12,v1r_c_ijp1k,v1r_c_ijk,v1r_c_ijm1k,&
             v1r_c_ijp1kp12,v1r_c_ijkp12,v1r_c_ijm1kp12,&
             v2r_ijp1k,v2r_ijk,v2r_ijm1k,v2r_ijp1kp12,&
             v2r_ijkp12,v2r_ijm1kp12,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 9 in main!                                      "
  END IF

  DEALLOCATE(v1r_ijkp12_new,v2r_ijkp12_new,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 10 in main!                                     "
  END IF

  DEALLOCATE(v1z_ip1jkp12,v1z_ijkp12,v1z_im1jkp12,v1z_ip1jkp1,v1z_ijkp1,&
             v1z_im1jkp1,v1z_c_ip1jkp12,v1z_c_ijkp12,v1z_c_im1jkp12,&
             v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
             v2z_ip1jkp12,v2z_ijkp12,v2z_im1jkp12,&
             v2z_ip1jkp1,v2z_ijkp1,v2z_im1jkp1,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 11 in main!                                     "
  END IF

  DEALLOCATE(v1z_ijkp1_new,v2z_ijkp1_new,stat=problem)
  IF (problem/=0) THEN
    PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
    PRINT *," Error code 12 in main!                                     "
  END IF

  PRINT *, " EXECUTION FINISHED! "
  STOP

END IF
```

```
! ============================================================================== !
! ============================================================================== !
! ============================================================================== !
dt  = dt_Plastic
tol = tol_Plastic

! "Lambda = -1" means that time dependent calculations (basically thixotropic)
!  have begun with time dependent shear viscosity ETA = ETA(SR,t,...).
Lambda = - 1.0D0

! Note that at the moment, then VELOCITY_k = VELOCITY_kp1_new.
PRINT *, "                                    "
PRINT *, " WRITING 't=0' INFORMATION TO FILE, STAND BY... "
CALL WRITE2FILE_k(VELOCITY_k,NX2)
CALL WRITE2FILE_kp1(VELOCITY_kp1_new,NX1,NY1,NX2,NY2,dr,dz)
CALL WRITE2FILE_torque_ZERO(VELOCITY_k,NX1,NX2,NY1,NY2,0,dt,Lambda,dr,dz,H3,omega)
PRINT *, " ...DONE! "

PRINT *, " NOW FINALLY, BEGINNING WITH THE TIME DEPENDENT PROBLEM... "

! ############################################################################## !
! In Section 7.11.2 is a detailed description of the algorithm, which is used in    !
! the following.                                                                     !
! ---------------------------------------------------------------------------------- !
! In this software, 'k' corresponds to t=(k+1/2)*dt and t=(k+1)*dt, and hence 'k = 0'
! means t = (1/2)*dt and t = 1*dt.
! This is due to the calling routine of the angular velocity:
! 'CALL ANGULAR_VELOCITY(DBLE(k)+1.0D0/2.0D0,dt,omega)' and
! 'CALL ANGULAR_VELOCITY(DBLE(k)+1.0D0,dt,omega)'.

! The fading memory modules are zero at k=0 (see Section 9.3.1):
FMSR = 0.0D0
FMCR = 0.0D0

N_dt = (NUMBER_OF_TIME_ITERATIONS - 1)
! ============================================================================== !
! ====================== Begin of the time loop (TIME) ========================= !
! ============================================================================== !
TIME_LOOP: DO k=0,N_dt

CONVERGENCE = .FALSE.

IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT *,"------------------------------------------------------------------------"
  PRINT *,"------------------------------------------------------------------------"
  PRINT *,"                                                                        "
  PRINT *,"  Calculating now for the time step: k+1/2 and k+1 = ",k+1," -> ->    "
  PRINT *,"  time = (k+1/2)dt = ",(DBLE(k)+1.0D0/2.0D0)*dt," SEC"
  PRINT *,"  time = (k+1)dt   = ",(DBLE(k)+1.0D0)*dt," SEC"
  PRINT *,"------------------------------------------------------------------------"
END IF
! ---------------------------------------------------------------------------------- !
! A guess for the time step k+1/2
VELOCITY_kp12 = VELOCITY_k

! Updating boundary condition at k+1/2:
! CALL ANGULAR_VELOCITY(k+1/2,dt,omega)
CALL ANGULAR_VELOCITY(DBLE(k)+1.0D0/2.0D0,dt,omega)

! Dirichlet boundary condition:
DO i = 1,NX2
  VELOCITY_kp12(i,1) = omega*DBLE(i-1)*dr
END DO
VELOCITY_kp12(1:NX1,NY1)       = 0.0D0
VELOCITY_kp12(1,1:NY1)         = 0.0D0
VELOCITY_kp12(NX1,NY1:NY2)     = 0.0D0
VELOCITY_kp12(NX2,:)           = R_o*omega

! Neumann boundary condition (this is actually redundant since the new
! information of "omega" cannot reach "VELOCITY_kp12(NX1+1:NX2-1,NY2-1)"
! or "VELOCITY_kp12(NX1+1:NX2-1,NY2-2)":
VELOCITY_kp12(NX1+1:NX2-1,NY2) = (4.0D0 * &
     VELOCITY_kp12(NX1+1:NX2-1,NY2-1) - &
     VELOCITY_kp12(NX1+1:NX2-1,NY2-2))/3.0D0
```

```
! Also updating boundary condition for "..._new":
VELOCITY_kp12_new = VELOCITY_kp12
! ------------------------------------------------------------------------------ !
! A guess for the time step k+1
VELOCITY_kp1 = VELOCITY_k

! Updating boundary condition at k+1:
! CALL ANGULAR_VELOCITY(k+1,dt,omega)
CALL ANGULAR_VELOCITY(DBLE(k)+1.0D0,dt,omega)

! Dirichlet boundary condition:
DO i = 1,NX2
  VELOCITY_kp1(i,1) = omega*DBLE(i-1)*dr
END DO
VELOCITY_kp1(1:NX1,NY1)       = 0.0D0
VELOCITY_kp1(1,1:NY1)         = 0.0D0
VELOCITY_kp1(NX1,NY1:NY2)     = 0.0D0
VELOCITY_kp1(NX2,:)           = R_o*omega

! Neumann boundary condition (this is actually redundant since the new
! information of "omega" cannot reach "VELOCITY_kp12(NX1+1:NX2-1,NY2-1)"
! or "VELOCITY_kp12(NX1+1:NX2-1,NY2-2)":
VELOCITY_kp1(NX1+1:NX2-1,NY2) = (4.0D0 * &
     VELOCITY_kp1(NX1+1:NX2-1,NY2-1) - &
     VELOCITY_kp1(NX1+1:NX2-1,NY2-2))/3.0D0

! Also updating boundary condition for "..._new":
VELOCITY_kp1_new = VELOCITY_kp1
! ------------------------------------------------------------------------------ !
count = 0
! ============================================================================== !
! ================== BEGIN OF SUCCESSIVE SUBSTITUTION (TIME) ==================== !
! ============================================================================== !
! The iteration loop here is because of the non-linearity of the governing
! Equations 7.22 and 7.23. To come around this problem, the successive substitution
! approach is used (see Section 7.8).
TIME_CONVERGE: DO WHILE (.NOT.CONVERGENCE)

! If convergence is a problem, then this might help:
! VELOCITY_kp12 = (VELOCITY_k + VELOCITY_kp1_new)/2
count = count + 1
12 FORMAT(4X,"Successive substitution (time) number = ",1(I3,1X))

IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT 12, count
END IF

!======================== BEGIN OF X-ITERATION (TIME) ========================
! Iteration is made along r-direction (i.e. along the i-direction as
! in A(i,j)). It starts at the bottom of the viscometer i=(2:NX2-1) at
! j = 2 and then move upward with increasing j (see Figures 8.1 and 8.2).

DO j = 2,NY1_cone-1
  v1r_ijp1k    = VELOCITY_k(:,j+1)
  v1r_ijk      = VELOCITY_k(:,j)
  v1r_ijm1k    = VELOCITY_k(:,j-1)
  v1r_ijp1kp12 = VELOCITY_kp12(:,j+1)
  v1r_ijkp12   = VELOCITY_kp12(:,j)
  v1r_ijm1kp12 = VELOCITY_kp12(:,j-1)

  FMSR1r_ijp1  = FMSR(:,j+1)
  FMSR1r_ij    = FMSR(:,j)
  FMSR1r_ijm1  = FMSR(:,j-1)

  FMCR1r_ijp1  = FMCR(:,j+1)
  FMCR1r_ij    = FMCR(:,j)
  FMCR1r_ijm1  = FMCR(:,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,0.0D0,dz,NX2,v1r_ijp1k,v1r_ijk,&
                       v1r_ijm1k,v1r_ijp1kp12,v1r_ijkp12,v1r_ijm1kp12,&
                       FMSR1r_ijp1,FMSR1r_ij,FMSR1r_ijm1,&
                       FMCR1r_ijp1,FMCR1r_ij,FMCR1r_ijm1,MX1,K1)
```

```
  CALL MATRIX_SOLVER(MX1,K1,v1r_ijkp12_new,NX2-2)
  VELOCITY_kp12_new(2:NX2-1,j) = v1r_ijkp12_new
END DO

! -------------------------- BEGIN OF X-CONE ------------------------
DO j = NY1_cone,NY1-1
  NX_cone                      = x_cone(j + 1 - NY1_cone)
  r_i_cone                     = NX_cone*dr
  NX2mNX_cone                  = NX2 - NX_cone
  v1r_c_ijp1k(1:NX2mNX_cone)    = VELOCITY_k(NX_cone+1:NX2,j+1)
  v1r_c_ijk(1:NX2mNX_cone)      = VELOCITY_k(NX_cone+1:NX2,j)
  v1r_c_ijm1k(1:NX2mNX_cone)    = VELOCITY_k(NX_cone+1:NX2,j-1)
  v1r_c_ijp1kp12(1:NX2mNX_cone) = VELOCITY_kp12(NX_cone+1:NX2,j+1)
  v1r_c_ijkp12(1:NX2mNX_cone)   = VELOCITY_kp12(NX_cone+1:NX2,j)
  v1r_c_ijm1kp12(1:NX2mNX_cone) = VELOCITY_kp12(NX_cone+1:NX2,j-1)

  FMSR1r_c_ijp1(1:NX2mNX_cone)  = FMSR(NX_cone+1:NX2,j+1)
  FMSR1r_c_ij(1:NX2mNX_cone)    = FMSR(NX_cone+1:NX2,j)
  FMSR1r_c_ijm1(1:NX2mNX_cone)  = FMSR(NX_cone+1:NX2,j-1)

  FMCR1r_c_ijp1(1:NX2mNX_cone)  = FMCR(NX_cone+1:NX2,j+1)
  FMCR1r_c_ij(1:NX2mNX_cone)    = FMCR(NX_cone+1:NX2,j)
  FMCR1r_c_ijm1(1:NX2mNX_cone)  = FMCR(NX_cone+1:NX2,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,r_i_cone,dz,NX2mNX_cone,v1r_c_ijp1k,v1r_c_ijk,&
                       v1r_c_ijm1k,v1r_c_ijp1kp12,v1r_c_ijkp12,v1r_c_ijm1kp12,&
                       FMSR1r_c_ijp1,FMSR1r_c_ij,FMSR1r_c_ijm1,&
                       FMCR1r_c_ijp1,FMCR1r_c_ij,FMCR1r_c_ijm1,MX1,K1)

  CALL MATRIX_SOLVER(MX1,K1,v1r_ijkp12_new,NX2mNX_cone-2)
  ! CALL WRITE2FILE_debug(MX1,K1,v1r_ijkp12_new,NX2mNX_cone-2)
  ! STOP
  VELOCITY_kp12_new(NX_cone+2:NX2-1,j) = v1r_ijkp12_new(1:NX2mNX_cone-2)
END DO
! -------------------------- END OF X-CONE -------------------------

DO j = NY1,NY2-1
  v2r_ijp1k    = VELOCITY_k(NX1:NX2,j+1)
  v2r_ijk      = VELOCITY_k(NX1:NX2,j)
  v2r_ijm1k    = VELOCITY_k(NX1:NX2,j-1)
  v2r_ijp1kp12 = VELOCITY_kp12(NX1:NX2,j+1)
  v2r_ijkp12   = VELOCITY_kp12(NX1:NX2,j)
  v2r_ijm1kp12 = VELOCITY_kp12(NX1:NX2,j-1)

  FMSR2r_ijp1  = FMSR(NX1:NX2,j+1)
  FMSR2r_ij    = FMSR(NX1:NX2,j)
  FMSR2r_ijm1  = FMSR(NX1:NX2,j-1)

  FMCR2r_ijp1  = FMCR(NX1:NX2,j+1)
  FMCR2r_ij    = FMCR(NX1:NX2,j)
  FMCR2r_ijm1  = FMCR(NX1:NX2,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,v2r_ijp1k,v2r_ijk,&
                       v2r_ijm1k,v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12,&
                       FMSR2r_ijp1,FMSR2r_ij,FMSR2r_ijm1,&
                       FMCR2r_ijp1,FMCR2r_ij,FMCR2r_ijm1,MX2,K2)

  CALL MATRIX_SOLVER(MX2,K2,v2r_ijkp12_new,NX-2)
  VELOCITY_kp12_new(NX1+1:NX2-1,j) = v2r_ijkp12_new
END DO

j = NY2
v2r_ijp1k    = VELOCITY_k(NX1:NX2,j-1)    ! => v(i,j+1) = v(i,j-1)
v2r_ijk      = VELOCITY_k(NX1:NX2,j)
v2r_ijm1k    = VELOCITY_k(NX1:NX2,j-1)
v2r_ijp1kp12 = VELOCITY_kp12(NX1:NX2,j-1) ! => v(i,j+1) = v(i,j-1)
v2r_ijkp12   = VELOCITY_kp12(NX1:NX2,j)
v2r_ijm1kp12 = VELOCITY_kp12(NX1:NX2,j-1)

FMSR2r_ijp1  = FMSR(NX1:NX2,j-1)          ! => v(i,j+1) = v(i,j-1)
FMSR2r_ij    = FMSR(NX1:NX2,j)
FMSR2r_ijm1  = FMSR(NX1:NX2,j-1)

FMCR2r_ijp1  = FMCR(NX1:NX2,j-1)          ! => v(i,j+1) = v(i,j-1)
```

```
FMCR2r_ij    = FMCR(NX1:NX2,j)
FMCR2r_ijm1  = FMCR(NX1:NX2,j-1)

CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,v2r_ijp1k,v2r_ijk,&
                     v2r_ijm1k,v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12,&
                     FMSR2r_ijp1,FMSR2r_ij,FMSR2r_ijm1,&
                     FMCR2r_ijp1,FMCR2r_ij,FMCR2r_ijm1,MX2,K2)

CALL MATRIX_SOLVER(MX2,K2,v2r_ijkp12_new,NX-2)
VELOCITY_kp12_new(NX1+1:NX2-1,j) = v2r_ijkp12_new

! --------- PAUSE FOR DEBUGGING ---------
! CALL WRITE2FILE_k(VELOCITY_kp12_new,NX2)
! WRITE(*,"(A)",ADVANCE="NO") " PRESS 'ENTER' TO CONTINUE "
! PRINT *, " "
! READ (*,"(A)") IGNORED_INPUT
! PRINT *, " "

!========================= END OF X-ITERATION (TIME) ========================

! Updating ..._kp12:
VELOCITY_kp12 = VELOCITY_kp12_new

!======================== BEGIN OF Y-ITERATION (TIME) ========================
! Iteration is made along z-direction (i.e. along the j-direction as
! in A(i,j)). It starts at the right side of the viscometer j=(2:NY2-1) at
! i = NX2-1 and then moves to the left with decreasing i
! (see Figures 8.1 and 8.2).

DO i = NX2-1,NX1+1,-1
  v2z_ip1jkp12 = VELOCITY_kp12(i+1,:)
  v2z_ijkp12   = VELOCITY_kp12(i,:)
  v2z_im1jkp12 = VELOCITY_kp12(i-1,:)
  v2z_ip1jkp1  = VELOCITY_kp1(i+1,:)
  v2z_ijkp1    = VELOCITY_kp1(i,:)
  v2z_im1jkp1  = VELOCITY_kp1(i-1,:)

  FMSR2z_ip1j  = FMSR(i+1,:)
  FMSR2z_ij    = FMSR(i,:)
  FMSR2z_im1j  = FMSR(i-1,:)

  FMCR2z_ip1j  = FMCR(i+1,:)
  FMCR2z_ij    = FMCR(i,:)
  FMCR2z_im1j  = FMCR(i-1,:)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY2,v2z_ip1jkp12,v2z_ijkp12,&
                       v2z_im1jkp12,v2z_ip1jkp1,v2z_ijkp1,v2z_im1jkp1,&
                       FMSR2z_ip1j,FMSR2z_ij,FMSR2z_im1j,&
                       FMCR2z_ip1j,FMCR2z_ij,FMCR2z_im1j,MY2,L2,.TRUE.)

  CALL MATRIX_SOLVER(MY2,L2,v2z_ijkp1_new,NY2-2)
  VELOCITY_kp1_new(i,2:NY2-1) = v2z_ijkp1_new
END DO

! v(i,j+1) = v(i,j-1) =>
VELOCITY_kp1_new(NX1+1:NX2-1,NY2) = VELOCITY_kp1_new(NX1+1:NX2-1,NY2-2)

DO i = NX1,NX2_cone+2,-1
  v1z_ip1jkp12 = VELOCITY_kp12(i+1,1:NY1)
  v1z_ijkp12   = VELOCITY_kp12(i,1:NY1)
  v1z_im1jkp12 = VELOCITY_kp12(i-1,1:NY1)
  v1z_ip1jkp1  = VELOCITY_kp1(i+1,1:NY1)
  v1z_ijkp1    = VELOCITY_kp1(i,1:NY1)
  v1z_im1jkp1  = VELOCITY_kp1(i-1,1:NY1)

  FMSR1z_ip1j  = FMSR(i+1,1:NY1)
  FMSR1z_ij    = FMSR(i,1:NY1)
  FMSR1z_im1j  = FMSR(i-1,1:NY1)

  FMCR1z_ip1j  = FMCR(i+1,1:NY1)
  FMCR1z_ij    = FMCR(i,1:NY1)
  FMCR1z_im1j  = FMCR(i-1,1:NY1)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY1,v1z_ip1jkp12,v1z_ijkp12,&
```

```
                      v1z_im1jkp12,v1z_ip1jkp1,v1z_ijkp1,v1z_im1jkp1,&
                      FMSR1z_ip1j,FMSR1z_ij,FMSR1z_im1j,&
                      FMCR1z_ip1j,FMCR1z_ij,FMCR1z_im1j,MY1,L1,.FALSE.)

  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY1-2)
  VELOCITY_kp1_new(i,2:NY1-1) = v1z_ijkp1_new
END DO

! -------------------------- BEGIN OF Y-CONE ------------------------
DO i = NX2_cone+1,NX1_cone+2,-1
  NY2_cone                    = y_cone(i-NX1_cone)
  v1z_c_ip1jkp12(1:NY2_cone)  = VELOCITY_kp12(i+1,1:NY2_cone)
  v1z_c_ijkp12(1:NY2_cone)    = VELOCITY_kp12(i,1:NY2_cone)
  v1z_c_im1jkp12(1:NY2_cone)  = VELOCITY_kp12(i-1,1:NY2_cone)
  v1z_c_ip1jkp1(1:NY2_cone)   = VELOCITY_kp1(i+1,1:NY2_cone)
  v1z_c_ijkp1(1:NY2_cone)     = VELOCITY_kp1(i,1:NY2_cone)
  v1z_c_im1jkp1(1:NY2_cone)   = VELOCITY_kp1(i-1,1:NY2_cone)

  FMSR1z_c_ip1j(1:NY2_cone)   = FMSR(i+1,1:NY2_cone)
  FMSR1z_c_ij(1:NY2_cone)     = FMSR(i,1:NY2_cone)
  FMSR1z_c_im1j(1:NY2_cone)   = FMSR(i-1,1:NY2_cone)

  FMCR1z_c_ip1j(1:NY2_cone)   = FMCR(i+1,1:NY2_cone)
  FMCR1z_c_ij(1:NY2_cone)     = FMCR(i,1:NY2_cone)
  FMCR1z_c_im1j(1:NY2_cone)   = FMCR(i-1,1:NY2_cone)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY2_cone,v1z_c_ip1jkp12,v1z_c_ijkp12,&
                      v1z_c_im1jkp12,v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
                      FMSR1z_c_ip1j,FMSR1z_c_ij,FMSR1z_c_im1j,&
                      FMCR1z_c_ip1j,FMCR1z_c_ij,FMCR1z_c_im1j,MY1,L1,.FALSE.)

  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY2_cone-2)
  VELOCITY_kp1_new(i,2:NY2_cone-1) = v1z_ijkp1_new(1:NY2_cone-2)
END DO

DO i = NX1_cone+1,2,-1
  v1z_c_ip1jkp12(1:NY1_cone)  = VELOCITY_kp12(i+1,1:NY1_cone)
  v1z_c_ijkp12(1:NY1_cone)    = VELOCITY_kp12(i,1:NY1_cone)
  v1z_c_im1jkp12(1:NY1_cone)  = VELOCITY_kp12(i-1,1:NY1_cone)
  v1z_c_ip1jkp1(1:NY1_cone)   = VELOCITY_kp1(i+1,1:NY1_cone)
  v1z_c_ijkp1(1:NY1_cone)     = VELOCITY_kp1(i,1:NY1_cone)
  v1z_c_im1jkp1(1:NY1_cone)   = VELOCITY_kp1(i-1,1:NY1_cone)

  FMSR1z_c_ip1j(1:NY1_cone)   = FMSR(i+1,1:NY1_cone)
  FMSR1z_c_ij(1:NY1_cone)     = FMSR(i,1:NY1_cone)
  FMSR1z_c_im1j(1:NY1_cone)   = FMSR(i-1,1:NY1_cone)

  FMCR1z_c_ip1j(1:NY1_cone)   = FMCR(i+1,1:NY1_cone)
  FMCR1z_c_ij(1:NY1_cone)     = FMCR(i,1:NY1_cone)
  FMCR1z_c_im1j(1:NY1_cone)   = FMCR(i-1,1:NY1_cone)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY1_cone,v1z_c_ip1jkp12,v1z_c_ijkp12,&
                      v1z_c_im1jkp12,v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
                      FMSR1z_c_ip1j,FMSR1z_c_ij,FMSR1z_c_im1j,&
                      FMCR1z_c_ip1j,FMCR1z_c_ij,FMCR1z_c_im1j,MY1,L1,.FALSE.)

  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY1_cone-2)
  VELOCITY_kp1_new(i,2:NY1_cone-1) = v1z_ijkp1_new(1:NY1_cone-2)
END DO
! -------------------------- END OF Y-CONE -------------------------

!======================== END OF Y-ITERATION (TIME) ========================

CONVERGENCE = .TRUE.

! Settings for testing of convergence (or rather stability):
RMS      = 0.0D0
vel_norm = 1.0D0

S5: DO j = 2,NY1_cone-1
      DO i = 2,NX2_cone+2
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
```

```
          CONVERGENCE = .FALSE.
          EXIT S5
        END IF
      END DO
    END DO S5

IF (CONVERGENCE) THEN
S6: DO j = NY1_cone,NY1-1
      NX1_cone_rms = x_cone(j-NY1_cone+1) + 2
      DO i = NX1_cone_rms,NX2_cone+2
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S6
        END IF
      END DO
    END DO S6
END IF

IF (CONVERGENCE) THEN
S7: DO j = 2,NY1-1
      DO i = NX2_cone+3,NX2-1
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S7
        END IF
      END DO
    END DO S7
END IF

IF (CONVERGENCE) THEN
S8: DO j = NY1,NY2
      DO i = NX1+1,NX2-1
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S8
        END IF
      END DO
    END DO S8
END IF

! Updating ..._kp1:
VELOCITY_kp1 = VELOCITY_kp1_new

IF (count == count_max) THEN
  CONVERGENCE       = .TRUE.
  FALSE_CONVERGENCE = .TRUE.
  PRINT *, " WARNING: FALSE CONVERGENCE! TIME STEP k= ", k
  PRINT *, " Maximum amount of successive substitutions is = ", count_max
  PRINT *, " ------------------------------------------------------------ "
  PRINT *, " RECOMMENDATION: Kill this application and reduce the time   "
  PRINT *, " step by an order of magnitude: dt -> dt/10                  "
  PRINT *, " ------------------------------------------------------------ "
END IF

END DO TIME_CONVERGE
! ============================================================================== !
! ==================== END OF SUCCESSIVE SUBSTITUTION (TIME) ===================== !
! ============================================================================== !
VELOCITY_k = VELOCITY_kp1_new
! ---------------------------------------------------------
CALL ROS_PROFILE(VELOCITY_k,NX1,NX2,NY1,NY2,DUMMY_1,dr,dz,SR,DUMMY_2)
time = (DBLE(k)+1.0D0)*dt
! ---------------------------------------------------------
DO i = 1,NX2
  DO j = 1,NY1
    SR_tmp = SR(i,j)
    CALL FMSR_FMCR(time,SR_tmp,U_o,memory_alpha,memory_beta,H_tmp)
    H(i,j) = H_tmp
```

```
  END DO
END DO
DO i = NX1,NX2
  DO j = NY1+1,NY2
    SR_tmp = SR(i,j)
    CALL FMSR_FMCR(time,SR_tmp,U_o,memory_alpha,memory_beta,H_tmp)
    H(i,j) = H_tmp
  END DO
END DO
! ---------------------------------------------------------
ALPHA_I = DEXP(time/memory_alpha)
FMSR    = ALPHA_I*SR*dt_Plastic + FMSR
! The above implicitly consists of two steps:
! 1) FMSR_kp1 = ALPHA_I*SR*dt_Plastic + FMSR_k (hence, time=(k+1)dt)
! 2) FMSR_k   = FMSR_kp1 <- Updating for the next time step,
!    just as the velocity was updated in the above =>
!    VELOCITY_k = VELOCITY_kp1_new (k <- k+1).
! ---------------------------------------------------------
BETA_I  = DEXP(time/memory_beta)
FMCR    = BETA_I*H*dt_Plastic + FMCR
! The above implicitly consists of two steps:
! 1) FMCR_kp1 = BETA_I*H*dt_Plastic + FMCR_k (hence, time=(k+1)dt)
! 2) FMCR_k   = FMCR_kp1 <- Updating for the next time step,
!    just as the velocity was updated in the above =>
!    VELOCITY_k = VELOCITY_kp1_new (k <- k+1).
! ------------------------------------------------------------------------------- !
! NOTE THAT THE VARIABLE "VELOCITY_k" NOW CONTAINS THE VELOCITY AT THE TIME
! STEP k+1, C.F. "VELOCITY_k = VELOCITY_kp1_new", JUST ABOVE!
! ---------------------------------------------------------
! Small output every 0.1 second (vel_corner.dat and vel_upper.dat):
! See Section 8.4.1 about vel_corner.dat and vel_upper.dat.
dt_OUTPUT_torque = 0.1D0
k_OUTPUT_torque  = IDNINT(dt_OUTPUT_torque/dt)
IF (ABS(MOD((k+1),k_OUTPUT_torque)).LT.small_zero) THEN
  PRINT *, " ================================================================ "
  PRINT *, " WRITING THE COMPACT DATA AT TIME ", DBLE(k+1)*dt
  PRINT *, " SECONDS TO FILE, STAND BY...     "
  CALL WRITE2FILE_torque(VELOCITY_k,FMSR,FMCR,NX1,NX2,NY1,NY2,k+1,dt,Lambda,dr,dz,H3,omega)
  PRINT *, " ...DONE! "
  PRINT *, " ================================================================ "
END IF

! Large output every 1 second (the whole domain of calculation -> vel_k.dat):
dt_OUTPUT = 1.0D0
k_OUTPUT  = IDNINT(dt_OUTPUT/dt) ! = 100000
IF (ABS(MOD((k+1),k_OUTPUT)).LT.small_zero) THEN
  PRINT *, " ================================================================ "
  PRINT *, " WRITING LARGE DATA AT TIME   ", DBLE(k+1)*dt
  PRINT *, " SECONDS TO FILE, STAND BY... "
  CALL WRITE2FILE_time(VELOCITY_k,NX2,k+1,dt)
  PRINT *, " ...DONE! "
  PRINT *, " ================================================================ "
END IF

END DO TIME_LOOP
! ============================================================================== !
! ======================== End of the time loop (TIME) ========================= !
! ============================================================================== !
IF (WARNING_SIGN) THEN
  PRINT *, " ---------------------------------------------------------- "
  PRINT *, " WARNING: For the time independent case, then:             "
  PRINT *, " k = MAX_NUMBER_OF_ITERATIONS; See log.dat                 "
  PRINT *, " ---------------------------------------------------------- "
END IF

IF (FALSE_CONVERGENCE) THEN
  PRINT *, " ---------------------------------------------------------- "
  PRINT *, " WARNING: For the time independent case, then:             "
  PRINT *, " FALSE CONVERGENCE WAS ACHIEVED. Reduce the time           "
  PRINT *, " step by order of magnitude: dt -> dt/10 and then rerun    "
  PRINT *, " the application.                                          "
  PRINT *, " ---------------------------------------------------------- "
END IF
```

```
! ------------------------------------------------------------------------------ !
! CLEARING SOME MAJOR VARIABLES FROM THE RANDOM ACCESS MEMORY:
DEALLOCATE(x_cone,y_cone,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 6 in main and execution terminated!             "
END IF

DEALLOCATE(MX1,MX2,MY1,MY2,VELOCITY_k,VELOCITY_kp12,VELOCITY_kp12_new,&
           VELOCITY_kp1,VELOCITY_kp1_new,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 7 in main!                                      "
END IF

DEALLOCATE(K1,K2,L1,L2,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 8 in main!                                      "
END IF

DEALLOCATE(v1r_ijp1k,v1r_ijk,v1r_ijm1k,v1r_ijp1kp12,v1r_ijkp12,&
           v1r_ijm1kp12,v1r_c_ijp1k,v1r_c_ijk,v1r_c_ijm1k,&
           v1r_c_ijp1kp12,v1r_c_ijkp12,v1r_c_ijm1kp12,&
           v2r_ijp1k,v2r_ijk,v2r_ijm1k,v2r_ijp1kp12,&
           v2r_ijkp12,v2r_ijm1kp12,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 9 in main!                                      "
END IF

DEALLOCATE(v1r_ijkp12_new,v2r_ijkp12_new,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 10 in main!                                     "
END IF

DEALLOCATE(v1z_ip1jkp12,v1z_ijkp12,v1z_im1jkp12,v1z_ip1jkp1,v1z_ijkp1,&
           v1z_im1jkp1,v1z_c_ip1jkp12,v1z_c_ijkp12,v1z_c_im1jkp12,&
           v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,&
           v2z_ip1jkp12,v2z_ijkp12,v2z_im1jkp12,&
           v2z_ip1jkp1,v2z_ijkp1,v2z_im1jkp1,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 11 in main!                                     "
END IF

DEALLOCATE(v1z_ijkp1_new,v2z_ijkp1_new,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 12 in main!                                     "
END IF

PRINT *, " Number of grid points (not including Dirichlet "
PRINT *, " boundary points) =   ", count_rms
PRINT *, "                      "
PRINT *, " EXECUTION FINISHED!  "
! ------------------------------------------------------------------------------ !
END PROGRAM MAIN_ROUTINE
! ------------------------------------------------------------------------------ !
```

A.2.6 shear.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: shear.f90 (MODULE)                                                  !
! This routine calculates the shear rate SR(i,j) from the computed velocity profile !
! VELOCITY_k(i,j). It is the program update.f90 that makes the request.          !
! See Section 7.5 about the formulas for the shear rate (SR). Note that ROS and SR  !
! means the same thing: ROS = rate of shear = SR = shear rate.                   !
! ------------------------------------------------------------------------------ !
MODULE RATE_OF_SHEAR
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: ROS
CONTAINS
! ============================================================================== !
SUBROUTINE ROS(rp1,r,rm1,dr,dz,V_ij,V_ip1j,V_im1j,V_ijp1,V_ijm1,V_ip1jp1,V_ip1jm1,&
               V_im1jp1,V_im1jm1,SR_ij,SR_ip12j,SR_im12j,SR_ijp12,SR_ijm12)

DOUBLE PRECISION,INTENT(IN)  ::  rp1,r,rm1,dr,dz,V_ij,V_ip1j,V_im1j,V_ijp1,&
                                 V_ijm1,V_ip1jp1,V_ip1jm1,V_im1jp1,V_im1jm1
DOUBLE PRECISION,INTENT(OUT) ::  SR_ij,SR_ip12j,SR_im12j,&
                                 SR_ijp12,SR_ijm12
DOUBLE PRECISION             ::  SR1_ij,SR2_ij,SR1_ip12j,SR2_ip12j,&
                                 SR1_im12j,SR2_im12j,SR1_ijp12,&
                                 SR2_ijp12,SR1_ijm12,SR2_ijm12
! ------------------------------------------------------------------------------ !
SR1_ij    = (V_ip1j - V_im1j)/(2.0D0*dr) - V_ij/r
SR2_ij    = (V_ijp1 - V_ijm1)/(2.0D0*dz)
SR_ij     = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)

SR1_ip12j = (V_ip1j - V_ij)/dr - (V_ip1j + V_ij)/(rp1 + r)
SR2_ip12j = (V_ip1jp1 + V_ijp1 - V_ip1jm1 - V_ijm1)/(4.0D0*dz)
SR_ip12j  = DSQRT(SR1_ip12j**2.0D0 + SR2_ip12j**2.0D0)

SR1_im12j = (V_ij - V_im1j)/dr - (V_ij + V_im1j)/(r + rm1)
SR2_im12j = (V_ijp1 + V_im1jp1 - V_ijm1 - V_im1jm1)/(4.0D0*dz)
SR_im12j  = DSQRT(SR1_im12j**2.0D0 + SR2_im12j**2.0D0)

SR1_ijp12 = (V_ip1jp1 + V_ip1j - V_im1jp1 - &
             V_im1j)/(4.0D0*dr) - (V_ijp1 + V_ij)/(2.0D0*r)
SR2_ijp12 = (V_ijp1 - V_ij)/dz
SR_ijp12  = DSQRT(SR1_ijp12**2.0D0 + SR2_ijp12**2.0D0)

SR1_ijm12 = (V_ip1j + V_ip1jm1 - V_im1j - &
             V_im1jm1)/(4.0D0*dr) - (V_ij + V_ijm1)/(2.0D0*r)
SR2_ijm12 = (V_ij - V_ijm1)/dz
SR_ijm12  = DSQRT(SR1_ijm12**2.0D0 + SR2_ijm12**2.0D0)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE ROS
! ============================================================================== !
END MODULE RATE_OF_SHEAR
```

A.2.7 motion.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: motion.f90 (MODULE)                                                 !
! This file reads the basic information from param.f90 to produce the angular    !
! velocity "omega". The information about the angular velocity is requested by the !
! routine main.f90.                                                              !
! ------------------------------------------------------------------------------ !
MODULE ROTATION
  USE CONSTANTS_AND_PARAMETERS
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: ANGULAR_VELOCITY
CONTAINS
! ============================================================================== !
SUBROUTINE ANGULAR_VELOCITY(double_prec_k,dt,omega)

DOUBLE PRECISION,INTENT(IN)  :: double_prec_k,dt
DOUBLE PRECISION,INTENT(OUT) :: omega

DOUBLE PRECISION             :: rho,REAL_TIME,ZERO_TIME,tol_Newton,&
                                tol_Plastic,tol_RMS,dt_Plastic,dt_Newton,&
                                R_i,R_o,h1,H2,H3
INTEGER                      :: count_max
LOGICAL                      :: ConTec_v4,ConTec_BML_v3,&
                                CALCULATE_TIME_DEPENDENT_PROBL
! ------------------------------------------------------------------------------ !
CALL WHAT_TYPE_OF_VISCOMETER(ConTec_v4,ConTec_BML_v3)
! ------------------------------------------------------
IF ((ConTec_v4).AND.(ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .TRUE.  "
  STOP
ELSE IF ((.NOT.ConTec_v4).AND.(.NOT.ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .FALSE. "
  STOP
END IF
! ------------------------------------------------------
IF (ConTec_v4) THEN
  CALL ConTec_CONSTANTS(rho,REAL_TIME,CALCULATE_TIME_DEPENDENT_PROBL,&
                        ZERO_TIME,tol_Newton,tol_Plastic,tol_RMS,&
                        dt_Plastic,dt_Newton,count_max,R_i,R_o,h1,H2,H3)
END IF

IF (ConTec_BML_v3) THEN
  CALL BML_CONSTANTS(rho,REAL_TIME,CALCULATE_TIME_DEPENDENT_PROBL,&
                     ZERO_TIME,tol_Newton,tol_Plastic,tol_RMS,&
                     dt_Plastic,dt_Newton,count_max,R_i,R_o,h1,H2,H3)
END IF
! ------------------------------------------------------
IF (CALCULATE_TIME_DEPENDENT_PROBL) THEN
  CALL STANDARD_OMEGA_UP_DOWN(double_prec_k,dt,omega)
ELSE
```

```
  CALL CONSTANT_OMEGA(omega)
END IF
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE ANGULAR_VELOCITY
! ============================================================================== !
SUBROUTINE CONSTANT_OMEGA(omega)

DOUBLE PRECISION,INTENT(OUT) :: omega

DOUBLE PRECISION             :: PI,ZERO_TIME,REAL_TIME,&
                                TIME_INTERVAL,f,f_min,f_max,PERC
INTEGER                      :: NUMBER_OF_POINTS
LOGICAL                      :: ConTec_v4,ConTec_BML_v3,SMOOTH
! ------------------------------------------------------------------------------ !
PI = DACOS(-1.0D0)
! ------------------------------------------------------
CALL WHAT_TYPE_OF_VISCOMETER(ConTec_v4,ConTec_BML_v3)
! ------------------------------------------------------
IF ((ConTec_v4).AND.(ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .TRUE.  "
  STOP
ELSE IF ((.NOT.ConTec_v4).AND.(.NOT.ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .FALSE. "
  STOP
END IF
! ------------------------------------------------------
IF (ConTec_v4) THEN
CALL VELOCITY_AND_TIME_ConTec(ZERO_TIME,REAL_TIME,&
     TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)
END IF

IF (ConTec_BML_v3) THEN
CALL VELOCITY_AND_TIME_BML(ZERO_TIME,REAL_TIME,&
     TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)
END IF
! ------------------------------------------------------
omega = 2.0D0*PI*f
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE CONSTANT_OMEGA
! ============================================================================== !
SUBROUTINE STANDARD_OMEGA_UP_DOWN(double_prec_k,dt,omega)

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:) :: omega_POINT,TIME_POINT,&
                                             omega_POINT_correction
INTEGER,ALLOCATABLE,DIMENSION(:)          :: OMEGA_STEP_VECTOR

DOUBLE PRECISION,INTENT(IN)  :: double_prec_k,dt
DOUBLE PRECISION,INTENT(OUT) :: omega

DOUBLE PRECISION             :: PI,f,f_min,f_max,OMEGA_MIN,OMEGA_MAX,&
                                TIME_INTERVAL,REAL_TIME,OMEGA_STEP,&
                                ZERO_TIME,PERC,EPS,time,SLOPE,step,&
                                value,value_1,value_2,value_3,value_4,&
                                f_start,f_end,OMEGA_start,OMEGA_end,&
                                PERC_static_begin,PERC_static_end

INTEGER                      :: NUMBER_OF_POINTS,NT_LOCK,problem,I,J,L

LOGICAL                      :: ConTec_v4,ConTec_BML_v3,SMOOTH
! ------------------------------------------------------------------------------ !
CALL WHAT_TYPE_OF_VISCOMETER(ConTec_v4,ConTec_BML_v3)
! ------------------------------------------------------
IF ((ConTec_v4).AND.(ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .TRUE.  "
  STOP
ELSE IF ((.NOT.ConTec_v4).AND.(.NOT.ConTec_BML_v3)) THEN
  PRINT *, " ERROR: Both ConTec_v4 AND ConTec_BML_v3 = .FALSE. "
  STOP
END IF
! ------------------------------------------------------
IF (ConTec_v4) THEN
CALL VELOCITY_AND_TIME_ConTec(ZERO_TIME,REAL_TIME,&
```

```
     TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)
END IF

IF (ConTec_BML_v3) THEN
CALL VELOCITY_AND_TIME_BML(ZERO_TIME,REAL_TIME,&
     TIME_INTERVAL,f,f_min,f_max,PERC,NUMBER_OF_POINTS,SMOOTH)
END IF
! ------------------------------------------------------------------------------ !
time   = double_prec_k*dt
PI     = DACOS(-1.0D0)
EPS    = 1.0D-9
! ------------------------------------------------------
! Parameters that should be ported into param.f90 in future:
f_start = 0.02D0              ! rps
f_end   = (7.8D0/11.7D0)*f_max ! rps

PERC_static_begin = 0.12D0
PERC_static_end   = 0.21D0
! ------------------------------------------------------------------------------ !
ALLOCATE(omega_POINT_correction(NUMBER_OF_POINTS+1),stat=problem)
IF (problem/=0) THEN
  PRINT *, " PROGRAM COULD NOT ALLOCATE SPACE FOR THE VECTOR "
  PRINT *, " omega_POINT_correction!                         "
  PRINT *, " EXECUTION TERMINATED!                           "
  STOP
END IF

! Here, omega_POINT_correction is in rps.
omega_POINT_correction = (/ -0.00061363240021D0,0.00012914263445D0,&
                             0.00138781741912D0,0.00249033048894D0,&
                             0.00323310552361D0,0.00266225011894D0,&
                             0.00138781741912D0,0.00030113269445D0,&
                            -0.00061363240021D0,0.00170215702079D0/)

! omega_POINT_correction = 0.0D0
! ------------------------------------------------------
IF (PERC_static_begin.LT.EPS) THEN
  f_start = f_min
END IF

! Converting omega_POINT_correction from rps to rad/s:
omega_POINT_correction = 2.0D0*PI*omega_POINT_correction

OMEGA_start = 2.0D0*PI*f_start ! rad/s
OMEGA_end   = 2.0D0*PI*f_end   ! rad/s
OMEGA_MIN   = 2.0D0*PI*f_min   ! rad/s
OMEGA_MAX   = 2.0D0*PI*f_max   ! rad/s
OMEGA_STEP  = (OMEGA_MAX - OMEGA_MIN)/(DBLE(NUMBER_OF_POINTS - 1)/2.0D0)
! ------------------------------------------------------------------------------ !
ALLOCATE(OMEGA_STEP_VECTOR(NUMBER_OF_POINTS),stat=problem)
IF (problem/=0) THEN
  PRINT *, " PROGRAM COULD NOT ALLOCATE SPACE FOR THE VECTOR "
  PRINT *, " OMEGA_STEP_VECTOR!                              "
  PRINT *, " EXECUTION TERMINATED!                           "
  STOP
END IF

OMEGA_STEP_VECTOR = 0

DO I=0,((NUMBER_OF_POINTS-1)/2),1
  OMEGA_STEP_VECTOR(I+1) = I
END DO

DO I=(((NUMBER_OF_POINTS-1)/2)-1),0,-1
  OMEGA_STEP_VECTOR(NUMBER_OF_POINTS-I) = I
END DO
! ------------------------------------------------------
ALLOCATE(omega_POINT(2*(NUMBER_OF_POINTS+1)+1),&
         TIME_POINT(2*(NUMBER_OF_POINTS+1)+1),stat=problem)
IF (problem/=0) THEN
  PRINT *, " PROGRAM COULD NOT ALLOCATE SPACE FOR THE VECTOR "
  PRINT *, " omega_POINT OR TIME_POINT!                      "
  PRINT *, " EXECUTION TERMINATED!                           "
  STOP
```

```
END IF

omega_POINT = 0.0D0
TIME_POINT  = 0.0D0
! ------------------------------------------------------------------------------ !
DO J=3,2*NUMBER_OF_POINTS-1,2 ! J=1,19,2
  TIME_POINT(J)   = DBLE((J+1)/2-1)*TIME_INTERVAL
  TIME_POINT(J+1) = TIME_POINT(J)+PERC*TIME_INTERVAL
  IF (J <= (NUMBER_OF_POINTS+1)) THEN
    omega_POINT(J)   = (OMEGA_MIN+OMEGA_STEP*&
                        DBLE(OMEGA_STEP_VECTOR((J+1)/2)-1))
    omega_POINT(J+1) = (OMEGA_MIN+OMEGA_STEP*&
                        DBLE(OMEGA_STEP_VECTOR((J+1)/2)))
  ELSE
    omega_POINT(J)   = (2.0D0*OMEGA_STEP+OMEGA_MIN+OMEGA_STEP*&
                        DBLE(OMEGA_STEP_VECTOR((J+1)/2)-1))
    omega_POINT(J+1) = (OMEGA_MIN+OMEGA_STEP*&
                        DBLE(OMEGA_STEP_VECTOR((J+1)/2)))
  END IF
END DO

J = 1
TIME_POINT(J)      = DBLE((J+1)/2-1)*TIME_INTERVAL
TIME_POINT(J+1)    = TIME_POINT(J) + PERC_static_begin*TIME_INTERVAL
omega_POINT(J)     = OMEGA_start
omega_POINT(J+1)   = OMEGA_MIN

J = 2*NUMBER_OF_POINTS + 1
TIME_POINT(J)      = DBLE((J+1)/2-1)*TIME_INTERVAL ! 45 sec
TIME_POINT(J+1)    = TIME_POINT(J) + PERC_static_end*TIME_INTERVAL
omega_POINT(J)     = OMEGA_MIN
omega_POINT(J+1)   = OMEGA_end

J = 2*NUMBER_OF_POINTS + 3
TIME_POINT(J)      = DBLE((J+1)/2-1)*TIME_INTERVAL
omega_POINT(J)     = OMEGA_end

DO J=2,2*NUMBER_OF_POINTS+2,2
  omega_POINT(J)   = omega_POINT(J)   + omega_POINT_correction(J/2)
  omega_POINT(J+1) = omega_POINT(J+1) + omega_POINT_correction(J/2)
END DO

! Legacy from debugging period (might come in use again):
! DO J=1,2*NUMBER_OF_POINTS+2
!   PRINT "( ' ', (F8.5,5X,F8.5) )  ",TIME_POINT(J),omega_POINT(J)
! END DO
! PRINT *, "time"
! READ *,time

DO L=1,2*NUMBER_OF_POINTS+2
  IF ((TIME_POINT(L).LE.time).AND.(TIME_POINT(L+1).GE.time)) THEN
    NT_LOCK=L
  END IF
END DO

L = NT_LOCK
! ------------------------------------------------------------------------------ !
! The following is to be used if smoothing is to be applied in only "one directions".
! EXP() - function is used.
! ------------------------------------------------------------
! When PERC = 0.17D0
value_1 = 5.8D0
value_2 = 7.0D0
value_3 = 6.5D0
value_4 = 6.0D0
! ------------------------------------------------------------
SLOPE = (omega_POINT(L+1) - omega_POINT(L))/(TIME_POINT(L+1) - TIME_POINT(L) + EPS)

OMEGA_STEP = DABS(omega_POINT(L+1) - omega_POINT(L))

IF (SMOOTH) THEN
! ------------------------------------------------------------
  IF (DABS(SLOPE).LT.1D-5) THEN
    omega = omega_POINT(L)
```

```
  ELSE
    IF (L.LT.2) THEN
      step  = (time - TIME_POINT(L))/(TIME_POINT(L+1) - TIME_POINT(L)) ! \in [0,1]
      value = (1.0D0 - DEXP(-step*value_1))                            ! \in [0,1]
      omega = OMEGA_STEP*value + omega_POINT(L)
    ELSE IF (L.LT.(NUMBER_OF_POINTS+1)) THEN !10
      step  = (time - TIME_POINT(L))/(TIME_POINT(L+1) - TIME_POINT(L)) ! \in [0,1]
      value = (1.0D0 - DEXP(-step*value_2))                            ! \in [0,1]
      omega = OMEGA_STEP*value + omega_POINT(L)
    ELSE IF (L.LT.(2*NUMBER_OF_POINTS)) THEN !18
      step  = (time - TIME_POINT(L))/(TIME_POINT(L+1) - TIME_POINT(L))
      value = - (1.0D0 - DEXP(-step*value_3))
      omega = OMEGA_STEP*value + omega_POINT(L)
    ELSE
      step  = (time - TIME_POINT(L))/(TIME_POINT(L+1) - TIME_POINT(L))
      value = (1.0D0 - DEXP(-step*value_4))
      omega = OMEGA_STEP*value + omega_POINT(L)
    END IF
  END IF
! ---------------------------------------------------------
ELSE
  omega = omega_POINT(L) + SLOPE*(time - TIME_POINT(L))
END IF
! ------------------------------------------------------------------------------- !
DEALLOCATE(OMEGA_STEP_VECTOR,stat=problem)
IF (problem/=0) THEN
  PRINT *, " PROGRAM COULD NOT DEALLOCATE SPACE FOR THE VECTOR "
  PRINT *, " OMEGA_STEP_VECTOR!                                "
END IF

DEALLOCATE(omega_POINT,omega_POINT_correction,TIME_POINT,stat=problem)
IF (problem/=0) THEN
  PRINT *, " PROGRAM COULD NOT DEALLOCATE SPACE FOR THE VECTOR "
  PRINT *, " omega_POINT OR TIME_POINT!                        "
END IF
! ------------------------------------------------------------------------------- !
RETURN
END SUBROUTINE STANDARD_OMEGA_UP_DOWN
! ============================================================================== !
END MODULE ROTATION
! ------------------------------------------------------------------------------- !
```

A.2.8 update.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: update.f90 (MODULE)                                                 !
! This file sets up the system of algebraic Equations 7.28 to 7.31. This file also  !
! contains the Thomas algorithm that is used in solving this system.             !
! ------------------------------------------------------------------------------ !
MODULE MATRIX
  USE RATE_OF_SHEAR
  USE SHEAR_VISCOSITY
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: MATRIX_UPDATE_X,MATRIX_UPDATE_Y,MATRIX_SOLVER
CONTAINS
! ============================================================================== !
SUBROUTINE MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,vr_ijp1k,vr_ijk,&
                           vr_ijm1k,vr_ijp1kp12,vr_ijkp12,vr_ijm1kp12,&
                           FMSR_ijp1_t,FMSR_ij_t,FMSR_ijm1_t,&
                           FMCR_ijp1_t,FMCR_ij_t,FMCR_ijm1_t,M,K_M)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(OUT) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)   :: K_M

DOUBLE PRECISION,DIMENSION(:),INTENT(IN)    :: vr_ijp1k,vr_ijk,&
                                               vr_ijm1k,vr_ijp1kp12,&
                                               vr_ijkp12,vr_ijm1kp12,&
                                               FMSR_ijp1_t,FMSR_ij_t,&
                                               FMSR_ijm1_t,&
                                               FMCR_ijp1_t,FMCR_ij_t,&
                                               FMCR_ijm1_t

DOUBLE PRECISION,INTENT(IN)                 :: dt,dr,R_i,dz,Lambda,rho
INTEGER,INTENT(IN)                          :: k,NX


DOUBLE PRECISION :: BETA,THETA,CHI,rp1,r,rm1

DOUBLE PRECISION :: V_ijk,V_ip1jk,V_im1jk,V_ijp1k,V_ijm1k,&
                    V_ip1jp1k,V_ip1jm1k,V_im1jp1k,V_im1jm1k,&
                    V_ijkp12,V_ip1jkp12,V_im1jkp12,V_ijp1kp12,V_ijm1kp12,&
                    V_ip1jp1kp12,V_ip1jm1kp12,V_im1jp1kp12,V_im1jm1kp12

DOUBLE PRECISION :: SR_ijk,SR_ip12jk,SR_im12jk,&
                    SR_ijp12k,SR_ijm12k,SR_ijkp12,&
                    SR_ip12jkp12,SR_im12jkp12,SR_ijp12kp12,&
                    SR_ijm12kp12

DOUBLE PRECISION :: FMSR_im1j,FMSR_ij,FMSR_ip1j,FMSR_im1jp1,FMSR_ijp1,FMSR_ip1jp1,&
                    FMSR_im1jm1,FMSR_ijm1,FMSR_ip1jm1,FMSR_ip12jp12,FMSR_ip12jm12,&
                    FMSR_im12jp12,FMSR_im12jm12,FMSR_ip12j,FMSR_im12j,&
                    FMSR_ijp12,FMSR_ijm12
```

```
DOUBLE PRECISION :: FMCR_im1j,FMCR_ij,FMCR_ip1j,FMCR_im1jp1,FMCR_ijp1,FMCR_ip1jp1,&
                    FMCR_im1jm1,FMCR_ijm1,FMCR_ip1jm1,FMCR_ip12jp12,FMCR_ip12jm12,&
                    FMCR_im12jp12,FMCR_im12jm12,FMCR_ip12j,FMCR_im12j,&
                    FMCR_ijp12,FMCR_ijm12

DOUBLE PRECISION :: ETA_ijk,ETA_ip12jk,ETA_im12jk,ETA_ijp12k,&
                    ETA_ijm12k,ETA_ijkp12,ETA_ip12jkp12,&
                    ETA_im12jkp12,ETA_ijp12kp12,ETA_ijm12kp12

DOUBLE PRECISION :: A_kp12,B_kp12,C_kp12,D_k,E_k,F_k,KK
DOUBLE PRECISION :: TIME_k,TIME_kp12
INTEGER          :: i
! ------------------------------------------------------------------------------ !
BETA      = dt/(2.0D0*dr*rho)
CHI       = dt/(2.0D0*dz*rho)
TIME_k    = DBLE(k)*dt
TIME_kp12 = (DBLE(k) + 0.5D0)*dt

M         =  0.0D0
K_M       =  0.0D0
! ------------------------------------------------------------------------------ !
! i = 1    => i = 2 in main.f90, i.e. near the center of the viscometer.
! i = NX-2 => j = NX2-1 in main.f90, i.e. near the outer cylinder.
DO i = 1,NX-2

  rp1   =  DBLE(i+1)*dr + R_i
  r     =  DBLE(i)*dr + R_i
  rm1   =  DBLE(i-1)*dr + R_i
  THETA =  dt/(r*rho)

  V_im1jk       = vr_ijk(i)
  V_ijk         = vr_ijk(i+1)
  V_ip1jk       = vr_ijk(i+2)
  V_im1jp1k     = vr_ijp1k(i)
  V_ijp1k       = vr_ijp1k(i+1)
  V_ip1jp1k     = vr_ijp1k(i+2)
  V_im1jm1k     = vr_ijm1k(i)
  V_ijm1k       = vr_ijm1k(i+1)
  V_ip1jm1k     = vr_ijm1k(i+2)

  V_im1jkp12    = vr_ijkp12(i)
  V_ijkp12      = vr_ijkp12(i+1)
  V_ip1jkp12    = vr_ijkp12(i+2)
  V_im1jp1kp12  = vr_ijp1kp12(i)
  V_ijp1kp12    = vr_ijp1kp12(i+1)
  V_ip1jp1kp12  = vr_ijp1kp12(i+2)
  V_im1jm1kp12  = vr_ijm1kp12(i)
  V_ijm1kp12    = vr_ijm1kp12(i+1)
  V_ip1jm1kp12  = vr_ijm1kp12(i+2)

  FMSR_im1j     = FMSR_ij_t(i)
  FMSR_ij       = FMSR_ij_t(i+1)
  FMSR_ip1j     = FMSR_ij_t(i+2)
  FMSR_im1jp1   = FMSR_ijp1_t(i)
  FMSR_ijp1     = FMSR_ijp1_t(i+1)
  FMSR_ip1jp1   = FMSR_ijp1_t(i+2)
  FMSR_im1jm1   = FMSR_ijm1_t(i)
  FMSR_ijm1     = FMSR_ijm1_t(i+1)
  FMSR_ip1jm1   = FMSR_ijm1_t(i+2)

  FMSR_ip12jp12 = (FMSR_ij   + FMSR_ip1j + FMSR_ip1jp1   + FMSR_ijp1)/4.0D0
  FMSR_ip12jm12 = (FMSR_ij   + FMSR_ip1j + FMSR_ip1jm1   + FMSR_ijm1)/4.0D0
  FMSR_im12jp12 = (FMSR_ij   + FMSR_im1j + FMSR_im1jp1   + FMSR_ijp1)/4.0D0
  FMSR_im12jm12 = (FMSR_ij   + FMSR_im1j + FMSR_im1jm1   + FMSR_ijm1)/4.0D0

  FMSR_ip12j    = (FMSR_ij   + FMSR_ip1j + FMSR_ip12jp12 + FMSR_ip12jm12)/4.0D0
  FMSR_im12j    = (FMSR_im1j + FMSR_ij   + FMSR_im12jp12 + FMSR_im12jm12)/4.0D0
  FMSR_ijp12    = (FMSR_ij   + FMSR_ijp1 + FMSR_ip12jp12 + FMSR_im12jp12)/4.0D0
  FMSR_ijm12    = (FMSR_ijm1 + FMSR_ij   + FMSR_ip12jm12 + FMSR_im12jm12)/4.0D0

  FMCR_im1j     = FMCR_ij_t(i)
  FMCR_ij       = FMCR_ij_t(i+1)
  FMCR_ip1j     = FMCR_ij_t(i+2)
  FMCR_im1jp1   = FMCR_ijp1_t(i)
```

```
FMCR_ijp1     = FMCR_ijp1_t(i+1)
FMCR_ip1jp1   = FMCR_ijp1_t(i+2)
FMCR_im1jm1   = FMCR_ijm1_t(i)
FMCR_ijm1     = FMCR_ijm1_t(i+1)
FMCR_ip1jm1   = FMCR_ijm1_t(i+2)

FMCR_ip12jp12 = (FMCR_ij   + FMCR_ip1j + FMCR_ip1jp1   + FMCR_ijp1)/4.0D0
FMCR_ip12jm12 = (FMCR_ij   + FMCR_ip1j + FMCR_ip1jm1   + FMCR_ijm1)/4.0D0
FMCR_im12jp12 = (FMCR_ij   + FMCR_im1j + FMCR_im1jp1   + FMCR_ijp1)/4.0D0
FMCR_im12jm12 = (FMCR_ij   + FMCR_im1j + FMCR_im1jm1   + FMCR_ijm1)/4.0D0

FMCR_ip12j    = (FMCR_ij   + FMCR_ip1j + FMCR_ip12jp12 + FMCR_ip12jm12)/4.0D0
FMCR_im12j    = (FMCR_im1j + FMCR_ij   + FMCR_im12jp12 + FMCR_im12jm12)/4.0D0
FMCR_ijp12    = (FMCR_ij   + FMCR_ijp1 + FMCR_ip12jp12 + FMCR_im12jp12)/4.0D0
FMCR_ijm12    = (FMCR_ijm1 + FMCR_ij   + FMCR_ip12jm12 + FMCR_im12jm12)/4.0D0

CALL ROS(rp1,r,rm1,dr,dz,V_ijk,V_ip1jk,V_im1jk,V_ijp1k,V_ijm1k,&
         V_ip1jp1k,V_ip1jm1k,V_im1jp1k,V_im1jm1k,&
         SR_ijk,SR_ip12jk,SR_im12jk,SR_ijp12k,SR_ijm12k)

CALL ROS(rp1,r,rm1,dr,dz,V_ijkp12,V_ip1jkp12,V_im1jkp12,V_ijp1kp12,V_ijm1kp12,&
         V_ip1jp1kp12,V_ip1jm1kp12,V_im1jp1kp12,V_im1jm1kp12,&
         SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,SR_ijp12kp12,SR_ijm12kp12)

! FMSR is relative to k and since we are calculating ETA_k, the "ALPHA_I*SR*dt"
! term must be set equal to zero! -> dt = 0.0D0.
! We use dt = 0.0D0 because we are calculating ETA for the time step k. I.e.
! we want to have Gamma = ALPHA_II*FMSR and not Gamma = ALPHA_II*(FMSR + ALPHA_I*SR*dt),
! because FMSR is already fully updated relative to the time step k.
! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
CALL ETA(0.0D0,TIME_k,Lambda,SR_ijk,SR_ip12jk,SR_im12jk,SR_ijp12k,SR_ijm12k,&
         FMSR_ij,FMSR_ip12j,FMSR_im12j,FMSR_ijp12,FMSR_ijm12,&
         FMCR_ij,FMCR_ip12j,FMCR_im12j,FMCR_ijp12,FMCR_ijm12,&
         ETA_ijk,ETA_ip12jk,ETA_im12jk,ETA_ijp12k,ETA_ijm12k)

! FMSR is relative to k and since we are calculating ETA_kp12, the "ALPHA_I*SR*dt"
! term must be set equal to "ALPHA_I*SR*(dt/2)"! -> dt = dt/2.
! FMSR is relative to k and we are only proceeding from k to k+1/2, hence dt -> dt/2.
! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
CALL ETA(dt/2.0D0,TIME_kp12,Lambda,SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,&
         SR_ijp12kp12,SR_ijm12kp12,FMSR_ij,FMSR_ip12j,FMSR_im12j,&
         FMSR_ijp12,FMSR_ijm12,FMCR_ij,FMCR_ip12j,FMCR_im12j,&
         FMCR_ijp12,FMCR_ijm12,ETA_ijkp12,ETA_ip12jkp12,&
         ETA_im12jkp12,ETA_ijp12kp12,ETA_ijm12kp12)

! Equations 7.32 to 7.34:
A_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr - 1.0D0/(rp1+r)) + (THETA*ETA_ijkp12)/(2.0D0*dr)
B_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr + 1.0D0/(rp1+r)) + &
         (BETA*ETA_im12jkp12)*(1.0D0/dr - 1.0D0/(r+rm1)) + (THETA*ETA_ijkp12)/r
C_kp12 = (BETA*ETA_im12jkp12)*(1.0D0/dr + 1.0D0/(r+rm1)) - (THETA*ETA_ijkp12)/(2.0D0*dr)

! Equations 7.35 to 7.37:
D_k    = (CHI*ETA_ijp12k)/dz
E_k    = (CHI*ETA_ijp12k)/dz + (CHI*ETA_ijm12k)/dz
F_k    = (CHI*ETA_ijm12k)/dz

! Equation 7.29:
KK = - D_k*V_ijp1k - (1.0D0 - E_k)*V_ijk - F_k*V_ijm1k

! Equation 7.28:
IF (i == 1) THEN
  M(i,i)   = -(1.0D0 + B_kp12)
  M(i,i+1) = A_kp12
  K_M(i)   = KK - C_kp12*V_im1jkp12
ELSE IF (i == NX-2) THEN
  M(i,i-1) = C_kp12
  M(i,i)   = -(1.0D0 + B_kp12)
  K_M(i)   = KK - A_kp12*V_ip1jkp12
ELSE
  M(i,i-1) = C_kp12
  M(i,i)   = -(1.0D0 + B_kp12)
  M(i,i+1) = A_kp12
  K_M(i)   = KK
END IF
```

```
END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE MATRIX_UPDATE_X
! ============================================================================== !
SUBROUTINE MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY,vz_ip1jkp12,vz_ijkp12,&
                           vz_im1jkp12,vz_ip1jkp1,vz_ijkp1,vz_im1jkp1,&
                           FMSR_ip1j_t,FMSR_ij_t,FMSR_im1j_t,&
                           FMCR_ip1j_t,FMCR_ij_t,FMCR_im1j_t,M,L,Neumann)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(OUT) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)   :: L

DOUBLE PRECISION,DIMENSION(:),INTENT(IN)    :: vz_im1jkp12,vz_ijkp12,&
                                               vz_ip1jkp12,vz_im1jkp1,&
                                               vz_ijkp1,vz_ip1jkp1,&
                                               FMSR_ip1j_t,FMSR_ij_t,&
                                               FMSR_im1j_t,&
                                               FMCR_ip1j_t,FMCR_ij_t,&
                                               FMCR_im1j_t

DOUBLE PRECISION,INTENT(IN)                 :: dt,dr,dz,Lambda,rho
INTEGER,INTENT(IN)                          :: k,NY,i

LOGICAL,INTENT(IN)                          :: Neumann

DOUBLE PRECISION :: BETA,THETA,CHI,rp1,r,rm1

DOUBLE PRECISION :: V_ijkp12,V_ip1jkp12,V_im1jkp12,V_ijp1kp12,V_ijm1kp12,&
                    V_ip1jp1kp12,V_ip1jm1kp12,V_im1jp1kp12,V_im1jm1kp12,&
                    V_ijkp1,V_ip1jkp1,V_im1jkp1,V_ijp1kp1,V_ijm1kp1,&
                    V_ip1jp1kp1,V_ip1jm1kp1,V_im1jp1kp1,V_im1jm1kp1

DOUBLE PRECISION :: SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,&
                    SR_ijp12kp12,SR_ijm12kp12,SR_ijkp1,&
                    SR_ip12jkp1,SR_im12jkp1,SR_ijp12kp1,&
                    SR_ijm12kp1

DOUBLE PRECISION :: FMSR_im1j,FMSR_ij,FMSR_ip1j,FMSR_im1jp1,FMSR_ijp1,FMSR_ip1jp1,&
                    FMSR_im1jm1,FMSR_ijm1,FMSR_ip1jm1,FMSR_ip12jp12,FMSR_ip12jm12,&
                    FMSR_im12jp12,FMSR_im12jm12,FMSR_ip12j,FMSR_im12j,&
                    FMSR_ijp12,FMSR_ijm12

DOUBLE PRECISION :: FMCR_im1j,FMCR_ij,FMCR_ip1j,FMCR_im1jp1,FMCR_ijp1,FMCR_ip1jp1,&
                    FMCR_im1jm1,FMCR_ijm1,FMCR_ip1jm1,FMCR_ip12jp12,FMCR_ip12jm12,&
                    FMCR_im12jp12,FMCR_im12jm12,FMCR_ip12j,FMCR_im12j,&
                    FMCR_ijp12,FMCR_ijm12

DOUBLE PRECISION :: ETA_ijkp12,ETA_ip12jkp12,ETA_im12jkp12,ETA_ijp12kp12,&
                    ETA_ijm12kp12,ETA_ijkp1,ETA_ip12jkp1,&
                    ETA_im12jkp1,ETA_ijp12kp1,ETA_ijm12kp1

DOUBLE PRECISION :: A_kp12,B_kp12,C_kp12,D_kp1,E_kp1,F_kp1,LL
DOUBLE PRECISION :: TIME_kp12,TIME_kp1
INTEGER          :: j
! ------------------------------------------------------------------------------ !
BETA      = dt/(2.0D0*dr*rho)
CHI       = dt/(2.0D0*dz*rho)
TIME_kp12 = (DBLE(k) + 0.5D0)*dt
TIME_kp1  = (DBLE(k) + 1.0D0)*dt

M         =  0.0D0
L         =  0.0D0

rp1       =  DBLE(i)*dr
r         =  DBLE(i-1)*dr
rm1       =  DBLE(i-2)*dr
THETA     =  dt/(r*rho)
! ------------------------------------------------------------------------------ !
! j = 1    => j = 2 in main.f90, i.e. near the bottom plate of the viscometer.
! j = NY-2 => j = NY2-1 in main.f90, i.e. near the top of the viscometer.
DO j = 1,NY-2
```

```
V_im1jkp12    = vz_im1jkp12(j+1)
V_ijkp12      = vz_ijkp12(j+1)
V_ip1jkp12    = vz_ip1jkp12(j+1)
V_im1jp1kp12  = vz_im1jkp12(j+2)
V_ijp1kp12    = vz_ijkp12(j+2)
V_ip1jp1kp12  = vz_ip1jkp12(j+2)
V_im1jm1kp12  = vz_im1jkp12(j)
V_ijm1kp12    = vz_ijkp12(j)
V_ip1jm1kp12  = vz_ip1jkp12(j)

V_im1jkp1     = vz_im1jkp1(j+1)
V_ijkp1       = vz_ijkp1(j+1)
V_ip1jkp1     = vz_ip1jkp1(j+1)
V_im1jp1kp1   = vz_im1jkp1(j+2)
V_ijp1kp1     = vz_ijkp1(j+2)
V_ip1jp1kp1   = vz_ip1jkp1(j+2)
V_im1jm1kp1   = vz_im1jkp1(j)
V_ijm1kp1     = vz_ijkp1(j)
V_ip1jm1kp1   = vz_ip1jkp1(j)

FMSR_im1j     = FMSR_im1j_t(j+1)
FMSR_ij       = FMSR_ij_t(j+1)
FMSR_ip1j     = FMSR_ip1j_t(j+1)
FMSR_im1jp1   = FMSR_im1j_t(j+2)
FMSR_ijp1     = FMSR_ij_t(j+2)
FMSR_ip1jp1   = FMSR_ip1j_t(j+2)
FMSR_im1jm1   = FMSR_im1j_t(j)
FMSR_ijm1     = FMSR_ij_t(j)
FMSR_ip1jm1   = FMSR_ip1j_t(j)

FMSR_ip12jp12 = (FMSR_ij   + FMSR_ip1j + FMSR_ip1jp1   + FMSR_ijp1)/4.0D0
FMSR_ip12jm12 = (FMSR_ij   + FMSR_ip1j + FMSR_ip1jm1   + FMSR_ijm1)/4.0D0
FMSR_im12jp12 = (FMSR_ij   + FMSR_im1j + FMSR_im1jp1   + FMSR_ijp1)/4.0D0
FMSR_im12jm12 = (FMSR_ij   + FMSR_im1j + FMSR_im1jm1   + FMSR_ijm1)/4.0D0

FMSR_ip12j    = (FMSR_ij   + FMSR_ip1j + FMSR_ip12jp12 + FMSR_ip12jm12)/4.0D0
FMSR_im12j    = (FMSR_im1j + FMSR_ij   + FMSR_im12jp12 + FMSR_im12jm12)/4.0D0
FMSR_ijp12    = (FMSR_ij   + FMSR_ijp1 + FMSR_ip12jp12 + FMSR_im12jp12)/4.0D0
FMSR_ijm12    = (FMSR_ijm1 + FMSR_ij   + FMSR_ip12jm12 + FMSR_im12jm12)/4.0D0

FMCR_im1j     = FMCR_im1j_t(j+1)
FMCR_ij       = FMCR_ij_t(j+1)
FMCR_ip1j     = FMCR_ip1j_t(j+1)
FMCR_im1jp1   = FMCR_im1j_t(j+2)
FMCR_ijp1     = FMCR_ij_t(j+2)
FMCR_ip1jp1   = FMCR_ip1j_t(j+2)
FMCR_im1jm1   = FMCR_im1j_t(j)
FMCR_ijm1     = FMCR_ij_t(j)
FMCR_ip1jm1   = FMCR_ip1j_t(j)

FMCR_ip12jp12 = (FMCR_ij   + FMCR_ip1j + FMCR_ip1jp1   + FMCR_ijp1)/4.0D0
FMCR_ip12jm12 = (FMCR_ij   + FMCR_ip1j + FMCR_ip1jm1   + FMCR_ijm1)/4.0D0
FMCR_im12jp12 = (FMCR_ij   + FMCR_im1j + FMCR_im1jp1   + FMCR_ijp1)/4.0D0
FMCR_im12jm12 = (FMCR_ij   + FMCR_im1j + FMCR_im1jm1   + FMCR_ijm1)/4.0D0

FMCR_ip12j    = (FMCR_ij   + FMCR_ip1j + FMCR_ip12jp12 + FMCR_ip12jm12)/4.0D0
FMCR_im12j    = (FMCR_im1j + FMCR_ij   + FMCR_im12jp12 + FMCR_im12jm12)/4.0D0
FMCR_ijp12    = (FMCR_ij   + FMCR_ijp1 + FMCR_ip12jp12 + FMCR_im12jp12)/4.0D0
FMCR_ijm12    = (FMCR_ijm1 + FMCR_ij   + FMCR_ip12jm12 + FMCR_im12jm12)/4.0D0

CALL ROS(rp1,r,rm1,dr,dz,V_ijkp12,V_ip1jkp12,V_im1jkp12,V_ijp1kp12,V_ijm1kp12,&
         V_ip1jp1kp12,V_ip1jm1kp12,V_im1jp1kp12,V_im1jm1kp12,&
         SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,SR_ijp12kp12,SR_ijm12kp12)

CALL ROS(rp1,r,rm1,dr,dz,V_ijkp1,V_ip1jkp1,V_im1jkp1,V_ijp1kp1,V_ijm1kp1,&
         V_ip1jp1kp1,V_ip1jm1kp1,V_im1jp1kp1,V_im1jm1kp1,&
         SR_ijkp1,SR_ip12jkp1,SR_im12jkp1,SR_ijp12kp1,SR_ijm12kp1)

! FMSR is relative to k and since we are calculating ETA_kp12, the "ALPHA_I*SR*dt"
! term must be set equal to "ALPHA_I*SR*(dt/2)"! -> dt = dt/2.
! FMSR is relative to k and we are only proceeding from k to k+1/2, hence dt -> dt/2.
! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
CALL ETA(dt/2.0D0,TIME_kp12,Lambda,SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,&
         SR_ijp12kp12,SR_ijm12kp12,FMSR_ij,FMSR_ip12j,FMSR_im12j,&
```

```
          FMSR_ijp12,FMSR_ijm12,FMCR_ij,FMCR_ip12j,FMCR_im12j,&
          FMCR_ijp12,FMCR_ijm12,ETA_ijkp12,ETA_ip12jkp12,&
          ETA_im12jkp12,ETA_ijp12kp12,ETA_ijm12kp12)

  ! FMSR is relative to k and since we are calculating ETA_kp1, the "ALPHA_I*SR*dt"
  ! term must be set equal to "ALPHA_I*SR*dt"! -> dt = dt.
  ! FMSR is relative to k and we are now proceeding from k to k+1, hence dt -> dt.
  ! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
  CALL ETA(dt,TIME_kp1,Lambda,SR_ijkp1,SR_ip12jkp1,SR_im12jkp1,SR_ijp12kp1,SR_ijm12kp1,&
          FMSR_ij,FMSR_ip12j,FMSR_im12j,FMSR_ijp12,FMSR_ijm12,&
          FMCR_ij,FMCR_ip12j,FMCR_im12j,FMCR_ijp12,FMCR_ijm12,&
          ETA_ijkp1,ETA_ip12jkp1,ETA_im12jkp1,ETA_ijp12kp1,ETA_ijm12kp1)

  ! Equations 7.32 to 7.34:
  A_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr - 1.0D0/(rp1+r)) + (THETA*ETA_ijkp12)/(2.0D0*dr)
  B_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr + 1.0D0/(rp1+r)) + &
           (BETA*ETA_im12jkp12)*(1.0D0/dr - 1.0D0/(r+rm1)) + (THETA*ETA_ijkp12)/r
  C_kp12 = (BETA*ETA_im12jkp12)*(1.0D0/dr + 1.0D0/(r+rm1)) - (THETA*ETA_ijkp12)/(2.0D0*dr)

  ! Equations 7.35 to 7.37:
  D_kp1  = (CHI*ETA_ijp12kp1)/dz
  E_kp1  = (CHI*ETA_ijp12kp1)/dz + (CHI*ETA_ijm12kp1)/dz
  F_kp1  = (CHI*ETA_ijm12kp1)/dz

  ! Equation 7.31:
  LL = - A_kp12*V_ip1jkp12 - (1.0D0 - B_kp12)*V_ijkp12 - C_kp12*V_im1jkp12

  ! Equation 7.30:
  IF (j == 1) THEN
    M(j,j)   = -(1.0D0 + E_kp1)
    M(j,j+1) = D_kp1
    L(j)     = LL - F_kp1*V_ijm1kp1
  ELSE IF ((j.EQ.NY-2).AND.Neumann) THEN ! (Equation 7.46)
    M(j,j-1) = 2.0D0*F_kp1
    M(j,j)   = -(1.0D0 + E_kp1)
    L(j)     = LL
  ELSE IF (j == NY-2) THEN
    M(j,j-1) = F_kp1
    M(j,j)   = -(1.0D0 + E_kp1)
    L(j)     = LL - D_kp1*V_ijp1kp1
  ELSE
    M(j,j-1) = F_kp1
    M(j,j)   = -(1.0D0 + E_kp1)
    M(j,j+1) = D_kp1
    L(j)     = LL
  END IF

END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE MATRIX_UPDATE_Y
! ============================================================================== !
SUBROUTINE MATRIX_SOLVER(M,D,v,dim)
! Subroutine that solves the trigonal system "M * v = D" with the
! Thomas algorithm (also known as the "Crout reduction for tridiagonal linear
! systems" - algorithm).
! M ->  Left side of the linear system (a tridiagonal array).
! D ->  Right side of the linear system (a vector).
! v ->  The variable to be solved: v = (M)^(-1) * D.

DOUBLE PRECISION,DIMENSION(:,:),INTENT(INOUT) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(INOUT)   :: D
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)     :: v
INTEGER,INTENT(IN)                            :: dim

DOUBLE PRECISION                              :: coef
INTEGER                                       :: i
! ------------------------------------------------------------------------------ !
v = -1.0D0
! ------------------------------------------------------------------------------ !
! Forward and then back substitution:
DO i = 1,dim-1
  coef       = M(i+1,(i-1)+1)/M(i,i)
  M(i+1,i+1) = M(i+1,i+1) - coef*M(i,i+1)
```

```
  D(i+1)     = D(i+1) - coef*D(i)
END DO
v(dim) = D(dim)/M(dim,dim)
DO i = dim-1,1,-1
  v(i) = (D(i) - M(i,i+1)*v(i+1))/M(i,i)
END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE MATRIX_SOLVER
! ============================================================================== !
END MODULE MATRIX
! ------------------------------------------------------------------------------ !
```

A.2.9 write2f.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: write2f.f90 (MODULE)                                                !
! This file takes care of writing all computed data into the different files.    !
! It is only the source main.f90 that makes such request.                       !
! ------------------------------------------------------------------------------ !
MODULE WRITE_INFORMATION
  USE SHEAR_VISCOSITY
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: WARNING_FOR_WRITING,WRITE2FILE_k,WRITE2FILE_kp1,WRITE2FILE_time,&
            WRITE2FILE_torque_ZERO,WRITE2FILE_torque,WRITE2FILE_debug,&
            WRITE2FILE_rms,ROS_PROFILE
CONTAINS
! ============================================================================== !
SUBROUTINE WARNING_FOR_WRITING(NY)
INTEGER,INTENT(IN) :: NY

IF (NY > 500) THEN
  PRINT *, " ERROR: NY2 > 500 ( NY2 = ",NY,")"
  PRINT *, " FORMAT STATEMENT IN THE FILE 'write2f.f90' IS TO SHORT:     "
  PRINT *, " ERROR -> 10 FORMAT(1X,500(F7.4,1X))                         "
  PRINT *, " PLEASE MAKE THE NECESSARY ADJUSTMENT IN ALL THE SUBROUTINES "
  PRINT *, " OF THIS FILE. TERMINAL ERROR!                               "
  STOP
END IF

RETURN
END SUBROUTINE WARNING_FOR_WRITING
! ============================================================================== !
SUBROUTINE WRITE2FILE_k(VELOCITY,NX)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN) :: VELOCITY
INTEGER,INTENT(IN)                         :: NX
INTEGER                                    :: problem,i
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F7.4,1X))
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vel_time.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_time.dat! "
  RETURN
ELSE
  DO i = 1,NX
    WRITE (unit=8,fmt=10) VELOCITY(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
```

```
! Creating file to log time, for the transient calculation:
OPEN(unit=8,file="time_k.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: time_k.dat! "
  RETURN
ELSE
  WRITE (unit=8,fmt=10) 0.0D0
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_k
! ============================================================================== !
SUBROUTINE WRITE2FILE_kp1(V,NX1,NY1,NX2,NY2,dr,dz)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN)  :: V
INTEGER,INTENT(IN)                          :: NX1,NY1,NX2,NY2
DOUBLE PRECISION,INTENT(IN)                 :: dr,dz

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: SR,ETA,von_Mises
DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: SR_r

DOUBLE PRECISION  :: ETA_tmp,SR_tmp,TIME,Lambda
INTEGER           :: problem,i,j,NX,NYt08
! ------------------------------------------------------------------------------ !
NYt08    = IDNINT(0.8D0*DBLE(NY2))
NX       = NX2 - NX1 + 1
! ------------------------------------------------------------------------------ !
ALLOCATE(SR(NX2,NY2),SR_r(NX),ETA(NX2,NY2),von_Mises(NX2,NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_kp1: The program could not allocate space! "
  PRINT *," Error code 1 in write2f and execution terminated!     "
  STOP
END IF

SR        = 0.0D0
ETA       = 0.0D0
von_Mises = 0.0D0

TIME      = 0.0D0
Lambda    = 1.0D0
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE AT ALL POINTS: SR
! Rate of shear at z = NYt08 as function of r: SR_r (not used here!)
CALL ROS_PROFILE(V,NX1,NX2,NY1,NY2,NYt08,dr,dz,SR,SR_r)
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR VISCOSITY AT ALL POINTS:
ETA_tmp = 0.0D0

DO i=1,NX2
  DO j=1,NY2
    SR_tmp   = SR(i,j)
    ! Using dt = 0.0D0 here, because there is no iteration forward in time. The objectives
    ! here is simply to write the last data (of time = k*dt, where k=0) to the hard drive.
    ! I.e. we want to have Gamma=ALPHA_II*(FMSR+ALPHA_I*SR*dt)=ALPHA_II*(0+ALPHA_I*SR*0)=0,
    ! because no shear rate history exists at k = 0 (FMSR is already fully updated at k=0).
    ! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
    CALL VISCOSITY(0.0D0,TIME,Lambda,SR_tmp,0.0D0,0.0D0,ETA_tmp)
    ETA(i,j) = ETA_tmp
  END DO
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING THE von Mises SHEAR STRESS AT ALL POINTS:
DO i=1,NX2
  DO j=1,NY2
    von_Mises(i,j) = SR(i,j)*ETA(i,j)
  END DO
END DO
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F10.4,1X))
16 FORMAT(1X,500(F14.4,1X))
! ------------------------------------------------------------------------------ !
```

```
OPEN(unit=8,file="vel_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=10) V(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="ROS_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: ROS_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=10) SR(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="ETA_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: ETA_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=16) ETA(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vonMises_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vonMises_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=10) von_Mises(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
DEALLOCATE(SR,SR_r,ETA,von_Mises,stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_kp1: The program could not deallocate space! "
  PRINT *," Error code 2 in write2f!                                 "
END IF
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_kp1
! ============================================================================== !
SUBROUTINE WRITE2FILE_time(VELOCITY,NX,kp1,dt)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN) :: VELOCITY
DOUBLE PRECISION,INTENT(IN)                :: dt
INTEGER,INTENT(IN)                         :: NX,kp1
INTEGER                                    :: problem,i
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F7.4,1X))
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vel_time.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: vel_time.dat! "
```

```
  RETURN
ELSE
  DO i = 1,NX
    WRITE (unit=8,fmt=10) VELOCITY(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="time_k.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: time_k.dat! "
  RETURN
ELSE
  WRITE (unit=8,fmt=10) DBLE(kp1)*dt
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_time
! ============================================================================== !
SUBROUTINE WRITE2FILE_rms(k,rms)

DOUBLE PRECISION,INTENT(IN) :: rms
INTEGER,INTENT(IN)          :: k
INTEGER                     :: problem
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="log.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: log.dat! "
  RETURN
ELSE
  WRITE (unit=8,fmt=*) k,rms
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_rms
! ============================================================================== !
SUBROUTINE WRITE2FILE_torque_ZERO(V,NX1,NX2,NY1,NY2,kp1,dt,Lambda,dr,dz,H3,omega)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN)  :: V
DOUBLE PRECISION,INTENT(IN)                 :: dt,Lambda,dr,dz,H3,omega
INTEGER,INTENT(IN)                          :: NX1,NX2,NY1,NY2,kp1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: SR,ETA
DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: SR_r,ETA_r,ROS_torque,ETA_torque,&
                                               TORQUE_ON_INNER_CYLINDER,&
                                               TORQUE_ON_OUTER_CYLINDER,&
                                               TORQUE_ON_BOTTOM_PLATE
DOUBLE PRECISION :: ETA_tmp,SR_tmp,&
                    TIME,TORQUE_sum_Ri,&
                    TORQUE_sum_work
INTEGER          :: NY2mH3,NYt08,problem,i,j,NX,NY,&
                    count_i,count_j,NY_write
! ------------------------------------------------------------------------------ !
NYt08    = IDNINT(0.8D0*DBLE(NY2))
NY2mH3   = NY2 - IDNINT(H3/dz)
NY_write = 50
NX       = NX2 - NX1 + 1    ! -> Same as in main.f90
NY       = NY2 - NY2mH3 + 1 ! -> Or equally, NY = IDNINT(H3/dz) + 1
TIME     = DBLE(kp1)*dt

IF (NY2mH3.LT.NY1+1) THEN
  PRINT *, "-----------------------------------------------"
  PRINT *, "  ERROR:                                       "
  PRINT *, "    write2f.f90 says: Something is wrong!      "
  PRINT *, "    NY2-H3/dz < NY1+1. Check out if H3         "
  PRINT *, "    (in the file param.f90) is correct!        "
  PRINT *, "-----------------------------------------------"
```

```
  STOP
END IF

ALLOCATE(SR(NX2,NY2),ETA(NX2,NY2),SR_r(NX),ETA_r(NX),ROS_torque(NY),&
         ETA_torque(NY),TORQUE_ON_INNER_CYLINDER(NY),&
         TORQUE_ON_OUTER_CYLINDER(NY2),TORQUE_ON_BOTTOM_PLATE(NX2),&
         stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_torque_ZERO: The program could not allocate space! "
  PRINT *," Error code 3 in write2f and execution terminated!             "
  STOP
END IF

SR         = 0.0D0
ETA        = 0.0D0
SR_r       = 0.0D0
ETA_r      = 0.0D0
ROS_torque = 0.0D0
ETA_torque = 0.0D0

TORQUE_ON_INNER_CYLINDER = 0.0D0
TORQUE_ON_OUTER_CYLINDER = 0.0D0
TORQUE_ON_BOTTOM_PLATE   = 0.0D0
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE AT ALL POINTS: SR
! Rate of shear at z = NYt08 as function of r: SR_r
CALL ROS_PROFILE(V,NX1,NX2,NY1,NY2,NYt08,dr,dz,SR,SR_r)
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR VISCOSITY AT ALL POINTS: ETA
! Shear viscosity at z = NYt08 as function of r: ETA_r
ETA_tmp = 0.0D0
DO i=1,NX2
  DO j=1,NY2
    SR_tmp   = SR(i,j)
    ! Using dt = 0.0D0 here, because there is no iteration forward in time. The objectives
    ! here is simply to write the last data (of time = k*dt, where k=0) to the hard drive.
    ! I.e. we want to have Gamma=ALPHA_II*(FMSR+ALPHA_I*SR*dt)=ALPHA_II*(0+ALPHA_I*SR*0)=0,
    ! because no shear rate history exists at k = 0 (FMSR is already fully updated at k=0).
    ! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
    CALL VISCOSITY(0.0D0,TIME,Lambda,SR_tmp,0.0D0,0.0D0,ETA_tmp)
    ETA(i,j) = ETA_tmp
  END DO
END DO
ETA_tmp = 0.0D0
DO i=1,NX
  SR_tmp   = SR_r(i)
  ! Using dt = 0.0D0 here, because there is no iteration forward in time. The objectives
  ! here is simply to write the last data (of time = k*dt, where k=0) to the hard drive.
  ! I.e. we want to have Gamma=ALPHA_II*(FMSR+ALPHA_I*SR*dt)=ALPHA_II*(0+ALPHA_I*SR*0)=0,
  ! because no shear rate history exists at k = 0 (FMSR is already fully updated at k=0).
  ! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
  CALL VISCOSITY(0.0D0,TIME,Lambda,SR_tmp,0.0D0,0.0D0,ETA_tmp)
  ETA_r(i) = ETA_tmp
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING TORQUE APPLIED ON THE INNER CYLINDER, ON THE OUTER CYLINDER
! AND ON THE BOTTOM PLATE, FROM THE TEST MATERIAL:
CALL TORQUE(V,ETA,NX1,NX2,NY1,NY2,NY2mH3,dr,dz,TORQUE_ON_INNER_CYLINDER,&
            TORQUE_ON_OUTER_CYLINDER,TORQUE_ON_BOTTOM_PLATE)
! ------------------------------------------------------------
! ... AND THEN SUMMARIZING ALL THE TORQUE ELEMENTS
! for all z in H3 (i.e. at where torque is measured),...
TORQUE_sum_Ri = 0.0D0
DO j=1,NY
  TORQUE_sum_Ri = TORQUE_ON_INNER_CYLINDER(j) + TORQUE_sum_Ri
END DO
! ------------------------------------------------------------
! ... AND FOR WORK CALCULATIONS:
TORQUE_sum_work = 0.0D0
DO j=1,NY2
  TORQUE_sum_work = TORQUE_ON_OUTER_CYLINDER(j) + TORQUE_sum_work
END DO
DO i=1,NX2
  TORQUE_sum_work = TORQUE_ON_BOTTOM_PLATE(i) + TORQUE_sum_work
```

```
END DO
! ------------------------------------------------------------------------------ !
! WRITING INFORMATION TO FILES:
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F9.5,1X))
12 FORMAT(1X,2(F9.5,1X))
14 FORMAT(1X,500(F5.3,1X))
16 FORMAT(1X,(F6.2,1X),(F8.5,1X),(F10.5,1X),(F11.5,1X))
18 FORMAT(1X,500(F7.3,1X))
! ------------------------------------------------------------------------------ !
! WRITING INNER AND OUTER CYLINDER TO FILE:
OPEN(unit=8,file="RiRo_dr.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: RiRo_dr.dat! "
ELSE
  WRITE (unit=8,fmt=12) DBLE(NX1-1)*dr,DBLE(NX2-1)*dr
  WRITE (unit=8,fmt=12) dr,dz
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING ANGULAR VELOCITY TO FILE:
OPEN(unit=8,file="time_omega.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: time_omega.dat! "
ELSE
  WRITE (unit=8,fmt=12) DBLE(kp1)*dt,omega
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING VELOCITY TO FILE:
OPEN(unit=8,file="velocity_r.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: velocity_r.dat! "
ELSE
  WRITE (unit=8,fmt=10) V(NX1:NX2,NYt08)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR RATE TO FILE:
OPEN(unit=8,file="ros_r.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: ros_r.dat! "
ELSE
  WRITE (unit=8,fmt=10) SR_r(:)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR VISCOSITY TO FILE:
OPEN(unit=8,file="eta_r.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: eta_r.dat! "
ELSE
  WRITE (unit=8,fmt=10) ETA_r(:)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING TORQUE TO FILE:
OPEN(unit=8,file="torque_z.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: torque_z.dat! "
ELSE
  WRITE (unit=8,fmt=10) TORQUE_ON_INNER_CYLINDER(:)
END IF
```

```
CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING TORQUE TO FILE:
OPEN(unit=8,file="torque.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: torque.dat! "
ELSE
  WRITE (unit=8,fmt=12) DBLE(kp1)*dt,TORQUE_sum_Ri
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING WORK DONE BY THE OUTER CYLINDER AND BOTTOM PLATE, TO FILE:
OPEN(unit=8,file="work.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: work.dat! "
ELSE
  WRITE (unit=8,fmt=16) DBLE(kp1)*dt,omega,-TORQUE_sum_work,-TORQUE_sum_work*omega
  ! See the comment in the SUBROUTINE TORQUE about why the term "TORQUE_sum_work"
  ! is written with a negative value into the file "work.dat".
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING VELOCITY OF CORNER TO FILE:
OPEN(unit=8,file="vel_corner.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_corner.dat! "
ELSE
!NX1_begin = NX1/2 + DNINT(0.5D0*DBLE(NX1/2))
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=14) V(i,1:NY_write+1)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING VELOCITY OF UPPER PART TO FILE:
OPEN(unit=8,file="vel_upper.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_upper.dat! "
ELSE
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=14) V(i,NY2-NY_write:NY2)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR RATE (SR) OF CORNER TO FILE:
OPEN(unit=8,file="ROS_corner.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: ROS_corner.dat! "
ELSE
!NX1_begin = NX1/2 + DNINT(0.5D0*DBLE(NX1/2))
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=18) SR(i,1:NY_write+1)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR RATE (SR) OF UPPER PART TO FILE:
OPEN(unit=8,file="ROS_upper.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: ROS_upper.dat! "
ELSE
```

```
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=18) SR(i,NY2-NY_write:NY2)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
DEALLOCATE(SR,ETA,SR_r,ETA_r,ROS_torque,ETA_torque,&
           TORQUE_ON_INNER_CYLINDER,TORQUE_ON_OUTER_CYLINDER,&
           TORQUE_ON_BOTTOM_PLATE,stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_torque_ZERO: The program could not deallocate space! "
  PRINT *," Error code 4 in write2f!                                         "
END IF
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_torque_ZERO
! ============================================================================== !
SUBROUTINE WRITE2FILE_torque(V,FMSR,FMCR,NX1,NX2,NY1,NY2,kp1,dt,Lambda,dr,dz,H3,omega)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN)  :: V,FMSR,FMCR
DOUBLE PRECISION,INTENT(IN)                 :: dt,Lambda,dr,dz,H3,omega
INTEGER,INTENT(IN)                          :: NX1,NX2,NY1,NY2,kp1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: SR,ETA
DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: SR_r,ETA_r,ROS_torque,ETA_torque,&
                                               TORQUE_ON_INNER_CYLINDER,&
                                               TORQUE_ON_OUTER_CYLINDER,&
                                               TORQUE_ON_BOTTOM_PLATE
DOUBLE PRECISION :: ETA_tmp,SR_tmp,TIME,&
                    TORQUE_sum_Ri,FMSR_tmp,&
                    FMCR_tmp,TORQUE_sum_work

INTEGER          :: NY2mH3,NYt08,problem,i,j,NX,NY,&
                    count_i,count_j,NY_write
! ------------------------------------------------------------------------------ !
NYt08    = IDNINT(0.8D0*DBLE(NY2))
NY2mH3   = NY2 - IDNINT(H3/dz)
NY_write = 50
NX       = NX2 - NX1 + 1    ! -> Same as in main.f90
NY       = NY2 - NY2mH3 + 1 ! -> Or equally, NY = IDNINT(H3/dz) + 1
TIME     = DBLE(kp1)*dt

ALLOCATE(SR(NX2,NY2),ETA(NX2,NY2),SR_r(NX),ETA_r(NX),ROS_torque(NY),&
         ETA_torque(NY),TORQUE_ON_INNER_CYLINDER(NY),&
         TORQUE_ON_OUTER_CYLINDER(NY2),TORQUE_ON_BOTTOM_PLATE(NX2),&
         stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_torque: The program could not allocate space! "
  PRINT *," Error code 5 in write2f and execution terminated!        "
  STOP
END IF

SR         = 0.0D0
ETA        = 0.0D0
SR_r       = 0.0D0
ETA_r      = 0.0D0
ROS_torque = 0.0D0
ETA_torque = 0.0D0

TORQUE_ON_INNER_CYLINDER = 0.0D0
TORQUE_ON_OUTER_CYLINDER = 0.0D0
TORQUE_ON_BOTTOM_PLATE   = 0.0D0
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE AT ALL POINTS: SR
! Rate of shear at z = NYt08 as function of r: SR_r
CALL ROS_PROFILE(V,NX1,NX2,NY1,NY2,NYt08,dr,dz,SR,SR_r)
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR VISCOSITY AT ALL POINTS: ETA
! Shear viscosity at z = NYt08 as function of r: ETA_r
ETA_tmp = 0.0D0
DO i=1,NX2
  DO j=1,NY2
    SR_tmp   = SR(i,j)
```

```
    FMSR_tmp = FMSR(i,j)
    FMCR_tmp = FMCR(i,j)
    ! Using dt = 0.0D0 here, because there is no iterating forward in time. The objectives
    ! here is simply to write the last data (of time = (k+1)*dt) to the hard drive.
    ! I.e. we want to have Gamma=ALPHA_II*FMSR and not Gamma=ALPHA_II*(FMSR+ALPHA_I*SR*dt),
    ! because FMSR is already fully updated relative to k+1.
    ! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
    CALL VISCOSITY(0.0D0,TIME,Lambda,SR_tmp,FMSR_tmp,FMCR_tmp,ETA_tmp)
    ETA(i,j) = ETA_tmp
  END DO
END DO
ETA_tmp = 0.0D0
DO i=1,NX
  SR_tmp   = SR_r(i)
  FMSR_tmp = FMSR((NX1-1)+i,NY2mH3)
  FMCR_tmp = FMCR((NX1-1)+i,NY2mH3)
  ! Using dt = 0.0D0 here, because there is no iterating forward in time. The objectives
  ! here is simply to write the last data (of time = (k+1)*dt) to the hard drive.
  ! I.e. we want to have Gamma=ALPHA_II*FMSR and not Gamma=ALPHA_II*(FMSR+ALPHA_I*SR*dt),
  ! because FMSR is already fully updated relative to k+1.
  ! [ The same consideration applies for FMCR, BETA_I, BETA_II and H(SR) ].
  CALL VISCOSITY(0.0D0,TIME,Lambda,SR_tmp,FMSR_tmp,FMCR_tmp,ETA_tmp)
  ETA_r(i) = ETA_tmp
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING TORQUE APPLIED ON THE INNER CYLINDER, ON THE OUTER CYLINDER
! AND ON THE BOTTOM PLATE, FROM THE TEST MATERIAL:
CALL TORQUE(V,ETA,NX1,NX2,NY1,NY2,NY2mH3,dr,dz,TORQUE_ON_INNER_CYLINDER,&
            TORQUE_ON_OUTER_CYLINDER,TORQUE_ON_BOTTOM_PLATE)
! ------------------------------------------------------
! ... AND THEN SUMMARIZING ALL THE TORQUE ELEMENTS
! for all z in H3 (i.e. at where torque is measured),...
TORQUE_sum_Ri = 0.0D0
DO j=1,NY
  TORQUE_sum_Ri = TORQUE_ON_INNER_CYLINDER(j) + TORQUE_sum_Ri
END DO
! ------------------------------------------------------
! ... AND FOR WORK CALCULATIONS:
TORQUE_sum_work = 0.0D0
DO j=1,NY2
  TORQUE_sum_work = TORQUE_ON_OUTER_CYLINDER(j) + TORQUE_sum_work
END DO
DO i=1,NX2
  TORQUE_sum_work = TORQUE_ON_BOTTOM_PLATE(i) + TORQUE_sum_work
END DO
! ------------------------------------------------------------------------------ !
! WRITING INFORMATION TO FILES:
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F9.5,1X))
12 FORMAT(1X,2(F9.5,1X))
14 FORMAT(1X,500(F5.3,1X))
16 FORMAT(1X,(F6.2,1X),(F8.5,1X),(F10.5,1X),(F11.5,1X))
18 FORMAT(1X,500(F7.3,1X))
! ------------------------------------------------------------------------------ !
! WRITING ANGULAR VELOCITY TO FILE:
OPEN(unit=8,file="time_omega.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: time_omega.dat! "
ELSE
  WRITE (unit=8,fmt=12) DBLE(kp1)*dt,omega
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING VELOCITY TO FILE:
OPEN(unit=8,file="velocity_r.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: velocity_r.dat! "
ELSE
  WRITE (unit=8,fmt=10) V(NX1:NX2,NYt08)
END IF
```

```
CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR RATE TO FILE:
OPEN(unit=8,file="ros_r.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: ros_r.dat! "
ELSE
  WRITE (unit=8,fmt=10) SR_r(:)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR VISCOSITY TO FILE:
OPEN(unit=8,file="eta_r.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: eta_r.dat! "
ELSE
  WRITE (unit=8,fmt=10) ETA_r(:)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING TORQUE TO FILE:
OPEN(unit=8,file="torque_z.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: torque_z.dat! "
ELSE
  WRITE (unit=8,fmt=10) TORQUE_ON_INNER_CYLINDER(:)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING TORQUE TO FILE:
OPEN(unit=8,file="torque.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: torque.dat! "
ELSE
  WRITE (unit=8,fmt=12) DBLE(kp1)*dt,TORQUE_sum_Ri
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING WORK DONE BY THE OUTER CYLINDER AND BOTTOM PLATE, TO FILE:
OPEN(unit=8,file="work.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: work.dat! "
ELSE
  WRITE (unit=8,fmt=16) DBLE(kp1)*dt,omega,-TORQUE_sum_work,-TORQUE_sum_work*omega
  ! See the comment in the SUBROUTINE TORQUE about why the term "TORQUE_sum_work"
  ! is written with a negative value into the file "work.dat".
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING VELOCITY OF CORNER TO FILE:
OPEN(unit=8,file="vel_corner.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: vel_corner.dat! "
ELSE
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=14) V(i,1:NY_write+1)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING VELOCITY OF UPPER PART TO FILE:
OPEN(unit=8,file="vel_upper.dat",status="old",action="write",&
```

```
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: vel_upper.dat! "
ELSE
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=14) V(i,NY2-NY_write:NY2)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR RATE (SR) OF CORNER TO FILE:
OPEN(unit=8,file="ROS_corner.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: ROS_corner.dat! "
ELSE
!NX1_begin = NX1/2 + DNINT(0.5D0*DBLE(NX1/2))
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=18) SR(i,1:NY_write+1)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
! WRITING SHEAR RATE (SR) OF UPPER PART TO FILE:
OPEN(unit=8,file="ROS_upper.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: ROS_upper.dat! "
ELSE
  DO i = NX2-50,NX2
    WRITE (unit=8,fmt=18) SR(i,NY2-NY_write:NY2)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
DEALLOCATE(SR,ETA,SR_r,ETA_r,ROS_torque,ETA_torque,&
           TORQUE_ON_INNER_CYLINDER,TORQUE_ON_OUTER_CYLINDER,&
           TORQUE_ON_BOTTOM_PLATE,stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_torque: The program could not deallocate space! "
  PRINT *," Error code 6 in write2f!                                   "
END IF
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_torque
! ============================================================================== !
SUBROUTINE WRITE2FILE_debug(M,K_M,v_new,N)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(IN)   :: K_M,v_new
INTEGER,INTENT(IN)                         :: N
INTEGER                                    :: problem,i
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,201(F10.4,1X))
11 FORMAT(1X,F10.4)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="MM_debug.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: MM_debug.dat! "
  RETURN
ELSE
  DO i = 1,N
    WRITE (unit=8,fmt=10) M(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="KK_debug.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
```

```
IF (problem/=0) THEN
  PRINT *," Could not create the file: KK_debug.dat! "
  RETURN
ELSE
  DO i = 1,N
    WRITE (unit=8,fmt=11) K_M(i)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vel_debug.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_debug.dat! "
  RETURN
ELSE
  DO i = 1,N
    WRITE (unit=8,fmt=11) v_new(i)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_debug
! ============================================================================== !
SUBROUTINE ROS_PROFILE(V,NX1,NX2,NY1,NY2,NYt08,dr,dz,SR,SR_r)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN)  :: V
DOUBLE PRECISION,DIMENSION(:,:),INTENT(OUT) :: SR
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)   :: SR_r
INTEGER,INTENT(IN)                          :: NX1,NY1,NX2,NY2,NYt08
DOUBLE PRECISION,INTENT(IN)                 :: dr,dz

INTEGER                                     :: i,j
DOUBLE PRECISION                            :: r,SR1_ij,SR2_ij,EPS
! ------------------------------------------------------------------------------ !
EPS = 1.0D-15
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE IN THE BULK:
! See Section 7.5 about the formulas for the shear rate (SR). Note that ROS and SR
! means the same thing: ROS = rate of shear = SR = shear rate.
! ------------------------------------------------------------------------------ !
DO i=2,NX2-1
  r = DBLE(i-1)*dr
  DO j=2,NY2-1
    SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
    SR2_ij  = (V(i,j+1) - V(i,j-1))/(2.0D0*dz)
    SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
  END DO
END DO
! ------------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE "LEFT" WALL (r=R_i):
! SR2_ij is actually zero since the Dirichlet boundary
! condition is not chancing with z!
i = NX1
r = DBLE(i-1)*dr
DO j=NY1+1,NY2-1
  SR1_ij  = (4.0D0*V(i+1,j) - V(i+2,j) - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (V(i,j+1) - V(i,j-1))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE "RIGHT" WALL (r=R_o):
! SR2_ij is actually zero since the Dirichlet boundary
! condition is not chancing with z!
i = NX2
r = DBLE(i-1)*dr
DO j=2,NY2-1
  SR1_ij  = (-4.0D0*V(i-1,j) + V(i-2,j) + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (V(i,j+1) - V(i,j-1))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
```

```
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE BOTTOM PLATE (z=0):
j = 1
DO i=2,NX2-1
  r = DBLE(i-1)*dr
  SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (4.0D0*V(i,j+1) - V(i,j+2) - 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE BELOW THE INNER CYLINDER (z=H3-H2):
j = NY1
DO i=NX1-50,NX1-1
  r = DBLE(i-1)*dr
  SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE AT THE OPEN BOUNDARY (z=H):
j = NY2
DO i=NX1+1,NX2-1
  r = DBLE(i-1)*dr
  SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE AT THE CENTER OF THE BOTTOM PLATE (r=0,z=0):
! This calculation is redundant since the rate of shear at the center line
! ($r=0 \forall z \in [0,H]$) is zero due to symmetry in the r-direction and
! due to the Dirichlet boundary condition $v_{\rm i,j}=0$ at the center line.
! i = 1
! j = 1
! r = DBLE(i-1)*dr
! SR1_ij  = (V(i+1,j) - V(i+1,j))/(2.0D0*dr) - V(i,j)/r ! due to symmetry
! SR2_ij  = ( 4.0D0*V(i,j+1) - V(i,j+2) - 3.0D0*V(i,j))/(2.0D0*dz)
! SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ------------
! Rather enforcing a zero rate of shear at the center line:
SR(1,1:NY2) = 0.0D0
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE ON THE LOWER RIGHT CORNER (r=R_o,z=0):
i = NX2
j = 1
r = DBLE(i-1)*dr
SR1_ij  = (-4.0D0*V(i-1,j) + V(i-2,j) + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = ( 4.0D0*V(i,j+1) - V(i,j+2) - 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE TOP RIGHT CORNER (r=R_o,z=H):
i = NX2
j = NY2
r = DBLE(i-1)*dr
SR1_ij  = (-4.0D0*V(i-1,j) + V(i-2,j) + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE TOP LEFT CORNER (r=R_i,z=H):
i = NX1
j = NY2
r = DBLE(i-1)*dr
SR1_ij  = ( 4.0D0*V(i+1,j) - V(i+2,j) - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE AT CORNER OF INNER CYLINDER (r=R_i,z=h_1+h_cone):
i = NX1
j = NY1
r = DBLE(i-1)*dr
SR1_ij  = ( 4.0D0*V(i+1,j) - V(i+2,j) - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ------------------------------------------------------------------------------ !
```

```
SR_r = SR(NX1:NX2,NYt08)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE ROS_PROFILE
! ============================================================================== !
SUBROUTINE TORQUE(V,ETA,NX1,NX2,NY1,NY2,NY2mH3,dr,dz,TORQUE_ON_INNER_CYLINDER,&
                  TORQUE_ON_OUTER_CYLINDER,TORQUE_ON_BOTTOM_PLATE)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN) :: V,ETA
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)  :: TORQUE_ON_INNER_CYLINDER,&
                                              TORQUE_ON_OUTER_CYLINDER,&
                                              TORQUE_ON_BOTTOM_PLATE
INTEGER,INTENT(IN)                         :: NX1,NY1,NX2,NY2,NY2mH3
DOUBLE PRECISION,INTENT(IN)                :: dr,dz

INTEGER          :: i,j,count_j
DOUBLE PRECISION :: shear_stress_theta_r_i,shear_stress_theta_r_o,&
                    shear_stress_theta_r,r,PI
! ------------------------------------------------------------------------------ !
PI = DACOS(-1.0D0)
! ------------------------------------------------------------------------------ !
! The elements "TORQUE_ON_OUTER_CYLINDER", "TORQUE_ON_BOTTOM_PLATE"
! and "TORQUE_ON_INNER_CYLINDER" are the torque applied ON the
! corresponding wall boundary FROM the test material. The first two
! elements "TORQUE_ON_OUTER_CYLINDER" and "TORQUE_ON_BOTTOM_PLATE" are
! used when generating the file "work.dat". There it is desired to
! gain the torque applied FROM the wall boundary ON the test material.
! As such, when writing data into the file "work.dat", both the torque
! elements will be written with an opposite sign (i.e. "+" -> "-").
! No corresponding considerations are needed for the toque element
! "TORQUE_ON_INNER_CYLINDER".
! ------------------------------------------------------------------------------ !
! CALCULATING THE TORQUE ON THE RIGHT WALL, FROM THE TEST MATERIAL (r=R_o):
i = NX2
r = DBLE(i-1)*dr
DO j=2,NY2-1
  shear_stress_theta_r_o      = - ETA(i,j)*((-4.0D0*V(i-1,j) + V(i-2,j) &
                                + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r)
  TORQUE_ON_OUTER_CYLINDER(j) = r*(shear_stress_theta_r_o*dz*(2*PI*r))
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING THE TORQUE ON TOP RIGHT WALL, FROM THE TEST MATERIAL (r=R_o;z=H):
i = NX2
j = NY2
r = DBLE(i-1)*dr
shear_stress_theta_r_o      = - ETA(i,j)*((-4.0D0*V(i-1,j) + V(i-2,j) &
                              + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r)
TORQUE_ON_OUTER_CYLINDER(j) = r*(shear_stress_theta_r_o*(dz/2)*(2*PI*r))
! ------------------------------------------------------------
! CALCULATING THE TORQUE ON BOTTOM RIGHT WALL, FROM THE TEST MATERIAL (r=R_o;z=0):
i = NX2
j = 1
r = DBLE(i-1)*dr
shear_stress_theta_r_o      = - ETA(i,j)*((-4.0D0*V(i-1,j) + V(i-2,j) &
                              + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r)
TORQUE_ON_OUTER_CYLINDER(j) = r*(shear_stress_theta_r_o*(dz/2)*(2*PI*r))
! ------------------------------------------------------------
! CALCULATING THE TORQUE ON THE BOTTOM PLATE, FROM THE TEST MATERIAL (z=0):
j = 1
DO i=2,NX2-1
  r = DBLE(i-1)*dr
  shear_stress_theta_r      = ETA(i,j)*((4.0D0*V(i,j+1) - V(i,j+2) &
                              - 3.0D0*V(i,j))/(2.0D0*dz))
  TORQUE_ON_BOTTOM_PLATE(i) = r*(shear_stress_theta_r*dr*(2*PI*r))
END DO
! ------------------------------------------------------------
! CALCULATING THE TORQUE ON THE BOTTOM PLATE - RIGHT CORNER, FROM THE
! TEST MATERIAL (r=R_o;z=0):
j = 1
i = NX2
r = DBLE(i-1)*dr
shear_stress_theta_r      = ETA(i,j)*((4.0D0*V(i,j+1) - V(i,j+2) &
                            - 3.0D0*V(i,j))/(2.0D0*dz))
TORQUE_ON_BOTTOM_PLATE(i) = r*(shear_stress_theta_r*(dr/2)*(2*PI*r))
```

```
! ------------------------------------------------------------------------------ !
! CALCULATING THE TORQUE ON THE LEFT WALL, FROM THE TEST MATERIAL (r=R_i):
! for all z in H3 (i.e. at where torque is measured):
i = NX1
r = DBLE(i-1)*dr
count_j = 0
DO j=NY2mH3,NY2
  count_j = count_j + 1
  shear_stress_theta_r_i = ETA(i,j)*((4.0D0*V(i+1,j) - V(i+2,j) &
                           - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r)
  IF ((j.EQ.NY2mH3).OR.(j.EQ.NY2)) THEN
    TORQUE_ON_INNER_CYLINDER(count_j) = r*(shear_stress_theta_r_i*(dz/2)*(2*PI*r))
  ELSE
    TORQUE_ON_INNER_CYLINDER(count_j) = r*(shear_stress_theta_r_i*dz*(2*PI*r))
  END IF
END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE TORQUE
! ============================================================================== !
END MODULE WRITE_INFORMATION
! ------------------------------------------------------------------------------ !
```

A.3 The C^3P^2-Geometry

The left illustration of Figure A.3 demonstrates the solution geometry and boundary condition used for the C^3P^2-geometry. The spacing of grid points in r- and z-direction are uniform and equal to $\Delta r = \Delta z = 2\,\text{mm}$. With this geometry, the corner of the top plate consist of a (part) cone-geometry. As such, the overall viscometric geometry, is actually a combination of three types of geometries, the **C**one, the **C**oaxial **C**ylinders and the **P**arallel **P**late type. Therefore its designation given is the CCCPP-geometry, or simply the C^3P^2-geometry.

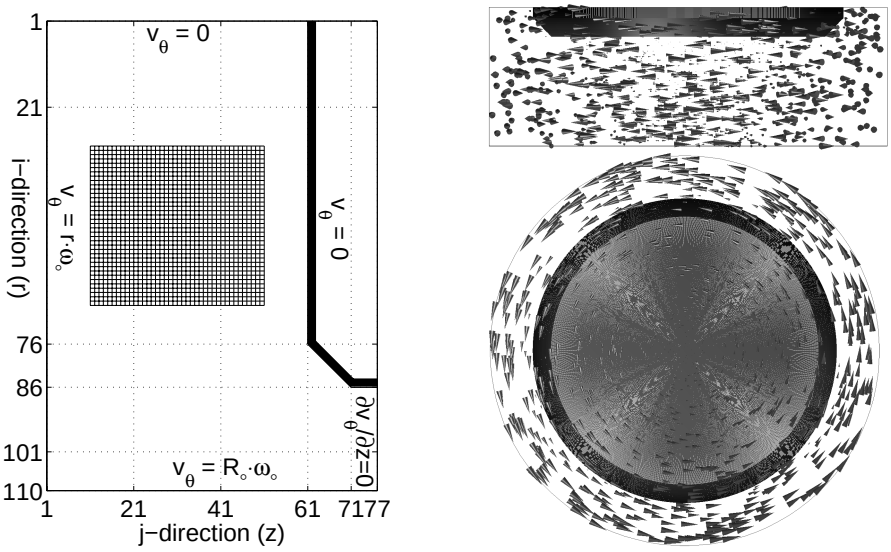


Figure A.3: To the left: Solution geometry and boundary condition for the C^3P^2-geometry. To the right: Vector plot of velocity $\mathbf{v}$, inside the C^3P^2-geometry (side and bottom view). The same condition applies here as in Figure 10.30, Page 264.

Referring to the left illustration of Figure A.3: In **i-direction** (i.e. in r-direction)

the following numbers specify the C^3P^2-geometry. The parameter NX1 = 76 designates the end coordinate of the top plate that measures torque (see discussion in Section 10.3.7). The term NX1_corner = 86 designates the radius of the inner cylinder R_i. The term NX2 = 110 designates the coordinate of the outer cylinder R_o. In the **j-direction** (i.e. in z-direction) the horizontal top plate begins at NY1_corner = 61. This point defines also the start location of the cone-geometry. This cone ends at NY1 = 71. The total height of the C^3P^2-geometry is defined with NY2 = 77. All the above number variables are used in the source code **main.f90**. With these numbers, the geometry in Figure A.3 can be read directly:

- $R_m = (\text{NX1} - 1)\,\Delta r = (76 - 1)\,2.0\,\text{mm} = 15.0\,\text{cm}$
- $R_i = (\text{NX1\_corner} - 1)\,\Delta r = (86 - 1)\,2.0\,\text{mm} = 17.0\,\text{cm}$
- $R_o = (\text{NX2} - 1)\,\Delta r = (110 - 1)\,2.0\,\text{mm} = 21.8\,\text{cm}$
- $h_{gap} = (\text{NY1\_corner} - 1)\,\Delta z = (61 - 1)\,2.0\,\text{mm} = 12.0\,\text{cm}$
- $h_{corner} = (\text{NY1} - \text{NY1\_corner})\,\Delta z = (71 - 61)\,2.0\,\text{mm} = 2.0\,\text{cm}$
- $h_{tot} = (\text{NY2} - 1)\,\Delta z = (77 - 1)\,2.0\,\text{mm} = 15.2\,\text{cm}$.

When calculating the viscoplastic flow inside the C^3P^2-Geometry, VISCOMETRIC-VISCOPLASTIC-FLOW consists of six files listed below. They are all very similar to what has already been shown in Appendix A.2, however somewhat simpler since time dependency is not of concern. That is, the following is based only on the algorithm shown in Section 7.11.1 (and not of Section 7.11.2).

1. **motion.f90** (MODULE): This file sets the (constant) angular velocity to be used $\omega_o = 2\,\pi\,f_o$ (i.e. ω_o = constant for each simulation).
2. **viscous.f90** (MODULE): In this file, the (time independent) shear viscosity function $\eta = \eta(\dot\gamma)$ is defined and calculated. This information is requested by the module **update.f90**.
3. **write2f.f90** (MODULE): This file takes care of writing all computed data into the different files. It is only the source **main.f90** that makes such request.
4. **shear.f90** (MODULE): This routine calculates the shear rate $\dot\gamma_c$ from the computed velocity profile V_k(i, j) $\mathbf{i}_\theta$. It is the program **update.f90** that makes the request.
5. **update.f90** (MODULE): This file sets up the system of algebraic Equations 7.28 to 7.31 (Page 161). This file also contains the Thomas algorithm that is used in solving this system.
6. **main.f90** (PROGRAM): This is the center of the whole software, holding and passing information to and from the different subroutines. Some subroutines interact directly with each other without going through the channels defined by **main.f90** (this applies mostly for the subroutines in the files **update.f90**, **shear.f90** and **viscous.f90**). The geometry of the viscometer is defined in this part of the software. This code also defines and sets all variables of relevance, like R_i, R_o, h, Δr, Δz, Δt, tol, tol_RMS and so forth.

All the files shown in the above list, can be compiled and linked in the order presented in the above. In the following, each source code is presented.

A.3.1 viscous.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: viscous.f90 (MODULE)                                                !
! In this file, the shear viscosity function ETA = ETA(SR) is defined and        !
! calculated. This information is requested by update.f90.                       !
! ------------------------------------------------------------------------------ !
MODULE SHEAR_VISCOSITY
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: ETA,VISCOSITY
CONTAINS
! ============================================================================== !
SUBROUTINE ETA(TIME,Lambda,ROS_ij,ROS_ip12j,ROS_im12j,ROS_ijp12,&
               ROS_ijm12,ETA_ij,ETA_ip12j,ETA_im12j,ETA_ijp12,ETA_ijm12)

DOUBLE PRECISION,INTENT(IN)    :: TIME,Lambda,ROS_ij,ROS_ip12j,ROS_im12j,&
                                  ROS_ijp12,ROS_ijm12

DOUBLE PRECISION,INTENT(OUT)   :: ETA_ij,ETA_ip12j,ETA_im12j,ETA_ijp12,&
                                  ETA_ijm12
! ------------------------------------------------------------------------------ !
CALL VISCOSITY(TIME,Lambda,ROS_ij,ETA_ij)
CALL VISCOSITY(TIME,Lambda,ROS_ip12j,ETA_ip12j)
CALL VISCOSITY(TIME,Lambda,ROS_im12j,ETA_im12j)
CALL VISCOSITY(TIME,Lambda,ROS_ijp12,ETA_ijp12)
CALL VISCOSITY(TIME,Lambda,ROS_ijm12,ETA_ijm12)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE ETA
! ============================================================================== !
SUBROUTINE VISCOSITY(TIME_tmp,Lambda_tmp,ROS,ETA)

DOUBLE PRECISION,INTENT(IN)    :: TIME_tmp,Lambda_tmp,ROS
DOUBLE PRECISION,INTENT(OUT)   :: ETA
DOUBLE PRECISION               :: mu,tau,delta,Lambda
! ------------------------------------------------------------------------------ !
! Lambda => Continuation Method (see Section 7.8).
Lambda = Lambda_tmp
! ------------------------------------------------------
mu     =  20.0D0  ! Plastic viscosity [Pa.s].
tau    = 150.0D0  ! Yield value [Pa].
delta  =  2.0D-3  ! <- The regularization parameter (see Section 7.9).
! ------------------------------------------------------
ETA = mu + (tau*Lambda)/(ROS + delta)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE VISCOSITY
! ============================================================================== !
END MODULE SHEAR_VISCOSITY
! ------------------------------------------------------------------------------ !
```

A.3.2 main.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: main.f90 (PROGRAM)                                                  !
! This is the center of the whole software, holding and passing information to and !
! from the different subroutines. Some subroutines interact directly with each   !
! other without going through the channels defined by main.f90 (this applies mostly !
! for the subroutines in the files update.f90, shear.f90 and viscous.f90).       !
! The geometry of the viscometer is defined in this part of the software.        !
! ------------------------------------------------------------------------------ !
PROGRAM MAIN_ROUTINE

USE ROTATION
USE MATRIX
USE WRITE_INFORMATION

IMPLICIT NONE

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: MX1,MX2,MY1,MY2
DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: K1,K2,L1,L2

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: VELOCITY_k,VELOCITY_kp12,&
                                               VELOCITY_kp12_new,&
                                               VELOCITY_kp1_new,VELOCITY_kp1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: &
                            v1r_ijp1k,      v1r_ijk,      v1r_ijm1k,&
                            v1r_ijp1kp12,   v1r_ijkp12,   v1r_ijm1kp12,&
                            v1r_c_ijp1k,    v1r_c_ijk,    v1r_c_ijm1k,&
                            v1r_c_ijp1kp12, v1r_c_ijkp12, v1r_c_ijm1kp12,&
                            v2r_ijp1k,      v2r_ijk,      v2r_ijm1k,&
                            v2r_ijp1kp12,   v2r_ijkp12,   v2r_ijm1kp12

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: v1r_ijkp12_new,v2r_ijkp12_new,&
                                               v1r_c_ijkp12_new

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: &
                            v1z_ip1jkp12,   v1z_ijkp12,   v1z_im1jkp12,&
                            v1z_ip1jkp1,    v1z_ijkp1,    v1z_im1jkp1,&
                            v1z_c_ip1jkp12, v1z_c_ijkp12, v1z_c_im1jkp12,&
                            v1z_c_ip1jkp1,  v1z_c_ijkp1,  v1z_c_im1jkp1,&
                            v2z_ip1jkp12,   v2z_ijkp12,   v2z_im1jkp12,&
                            v2z_ip1jkp1,    v2z_ijkp1,    v2z_im1jkp1

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: v1z_ijkp1_new,v2z_ijkp1_new,&
                                               v1z_c_ijkp1_new

INTEGER,ALLOCATABLE,DIMENSION(:)            :: x_corner,y_corner


DOUBLE PRECISION :: dr,dz,dt,dt_Plastic,dt_Newton,ZERO_TIME,rho,omega,R_i,&
                    R_i_corner,R_o,h_gap,h_corner,h_R_i,tol,tol_Newton,tol_Plastic,&
```

```
                   tol_RMS,tol_RMS_active,RMS,vel_norm,small_zero,EPS,a,b,Lambda

INTEGER          :: NX,NX_corner,NX1,NX1_corner,NX2,NX2mNX_corner,NY_corner,NY1,&
                    NY1_corner,NY2,NXY_corner,i,j,k,k_OUTPUT_rms,N_Lambda_MAX,&
                    N_Lambda,MAX_NUMBER_OF_ITERATIONS,problem,count,count_max,count_rms

LOGICAL          :: CONVERGENCE,TIME_INDEPENDENCE,WARNING_SIGN,FALSE_CONVERGENCE

CHARACTER        :: IGNORED_INPUT
! ------------------------------------------------------------------------------ !
PRINT *, "      ------------------------------------------          "
PRINT *, "        Viscometric-ViscoPlastic-Flow v1.0  (C3P2)        "
PRINT *, "                                                          "
PRINT *, "             Copyright (C) 2002, Jon E. Wallevik,         "
PRINT *, "                 (jon.wallevik@bygg.ntnu.no)              "
PRINT *, " The Norwegian University of Science and Technology (NTNU)  "
PRINT *, "             Department of Structural Engineering         "
PRINT *, " ---------------------------------------------------------- "
PRINT *, "                                                          "
PRINT *, " This software is free software; you can redistribute it    "
PRINT *, " and/or modify it under the terms of the GNU General Public "
PRINT *, " License as published by the Free Software Foundation;      "
PRINT *, " either version 2 of the License, or (at your option) any   "
PRINT *, " later version. This software is distributed in the hope    "
PRINT *, " that it will be useful, but WITHOUT ANY WARRANTY; without  "
PRINT *, " even the implied warranty of MERCHANTABILITY or FITNESS    "
PRINT *, " FOR A PARTICULAR PURPOSE.                                  "
PRINT *, " See the GNU General Public License for more details.       "
PRINT *, " ---------------------------------------------------------- "
PRINT *, "                                                          "
WRITE(*,"(A)",ADVANCE="NO") "               PRESS 'ENTER' TO CONTINUE"
PRINT *, " "
READ (*,"(A)") IGNORED_INPUT
PRINT *, " "
! ------------------------------------------------------------------------------ !
PRINT '(4X,A49)', "================================================="
PRINT '(4X,A49)', "          Solving for the C3P2-Geometry          "
PRINT '(4X,A49)', "================================================="
PRINT '(4X,A49)', "                                                 "
! ------------------------------------------------------------------------------ !
! ------------------------ BEGIN OF VARIABLE DECLARATION ----------------------- !
! ------------------------------------------------------------------------------ !
rho = 2354D0 ! kg/m^3 ! <- Note that for a time independent calculations, then
                      !    the value of rho is not physically important (i.e.
                      !    mass inertia plays no role).
! ------------
tol_Newton  = 1.0D-3  ! Used as condition for time independence in the Newtonian case.
                      ! Also used as tolerance for the successive substitution
                      ! (in this case it acts as a dummy variable since always two
                      ! successive steps are made for the Newtonian case).
tol_Plastic = 1.0D-10 ! For the successive substitution tolerance (Equation 7.73).
tol_RMS     = 1.0D-30 ! Condition for time independence (Equation 7.75).
! ------------
dt_Newton   = 1.0D-1
dt_Plastic  = 1.0D-5
count_max   = 15      ! Maximum number of successive (substitution) iterations,
                      ! for each (pseudotransient) time step k.
R_i         = 0.170D0 ! => 17.0 cm  = Inner radius of viscometer.
R_o         = 0.218D0 ! => 21.8 cm  = Outer radius of viscometer.
h_gap       = 12.0D-2 ! The gap between the parallel plates -> 12 cm.
h_R_i       = 12.0D-3 ! Top edge = 12 mm -> 12.0D-3
h_corner    =  2.0D-2 ! Size of corner in z-direction (for example 2.0D-2 -> 2 cm).
dr          =  2.0D-3 ! => 2.0 mm  = Spacing between grid points in r-direction.
dz          =  2.0D-3 ! => - " -   = Spacing between grid points in z-direction.
! ------------
ZERO_TIME = 4.00D0
k_OUTPUT_rms = 100    ! -> Information output every dt_OUTPUT_rms times (to console and file).
! ------------
! The term "small_zero" does usually not have to be changed.
small_zero = 0.1D-7   ! -> Used in relation to screen and file output.
EPS        = 1.0D-15  ! -> Used in relation to vel_norm.
! ------------------------------------------------------------------------------ !
! ------------------------- END OF VARIABLE DECLARATION ------------------------ !
! ------------------------------------------------------------------------------ !
```

```
NXY_corner  = IDNINT(h_corner/dz)              ! -> 2.0/0.2      = 10
NX1_corner  = IDNINT(R_i/dr) + 1               ! -> 17.0/0.2 + 1 = 86
NX1         = NX1_corner - NXY_corner          ! -> 86 - 10      = 76
NX2         = IDNINT(R_o/dr) + 1               ! -> 21.8/0.2 + 1 = 110
NX          = NX2 - NX1_corner + 1             ! -> 110 - 86 + 1 = 25
NY1_corner  = IDNINT(h_gap/dz) + 1             ! -> 12.0/0.2 + 1 = 61
NY1         = NY1_corner + NXY_corner          ! -> 61 + 10      = 71
NY2         = NY1 + IDNINT(h_R_i/dz)           ! -> 71 + 4       = 75
R_i         = dr*DBLE(NX1_corner-1)
R_o         = dr*DBLE(NX2-1)
! ------------------------------------------------------------------------------- !
CALL ANGULAR_VELOCITY(0.0D0,dt,omega)
! ------------------------------------------------------------------------------- !
ALLOCATE(x_corner(NY1-NY1_corner),y_corner(NX1_corner-NX1),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 0 in main and execution terminated!           "
  STOP
END IF
! ------------------------------------------------------------------------------- !
! x_corner = NX1_corner - (/11,10,9,8,7,6,5,4,3,2/)
DO j = NY1_corner,NY1-1
  k = j - NY1_corner + 1 ! k=1,10
  x_corner(k) = (NX1_corner - NX1 + 2) - k
! x_corner    = (/11,10,9,8,7,6,5,4,3,2/)
END DO
x_corner = NX1_corner - x_corner
DO i = NX1,NX1_corner-1
  k = i - NX1 + 1 ! k=1,10
  y_corner(k) = (NY1 - NY1_corner + 1) - k
END DO
y_corner = NY1_corner + y_corner
! ------------------------------------------------------------------------------- !
12 FORMAT(8X,"NX1 = ",(I3,1X),"; NX1_corner = ",(I3,1X),"; NX2 = ",(I3,1X))
13 FORMAT(8X,"NY1 = ",(I3,1X),"; NY1_corner = ",(I3,1X),"; NY2 = ",(I3,1X))
14 FORMAT(8X,"R_i = ",F6.4,"m ; R_o = ",F6.4,"m")
15 FORMAT(8X,"NX = ",(I3,1X),"; dr = ",F7.5,"m","; dz = ",F7.5,"m")
16 FORMAT(8X,"h_gap = ",F6.4,"m ; h_R_i = ",F6.4,"m")
17 FORMAT(8X,"h_corner = ",F6.4,"m")
18 FORMAT(8X,"dt_Plastic = ",E9.3,"s; f_o = ",F6.4,"rps")
PRINT '(7X,A26)',"Geometric and time values:"
PRINT 12, NX1,NX1_corner,NX2
PRINT 13, NY1,NY1_corner,NY2
PRINT 14, R_i,R_o
PRINT 15, NX,dr,dz
PRINT 16, dz*(NY1_corner-1),dz*(NY2-NY1)
PRINT 17, dz*(NY1 - NY1_corner)
PRINT 18, dt_Plastic,omega/(2.0D0*ACOS(-1.0D0))
PRINT *, "    "
! ------------------------------------------------------------------------------- !
IF (NX1_corner.GE.NX2) THEN
  PRINT *, " Inner radius 'R_i' is larger or equal to the outer radius 'R_o'! "
  PRINT *, " TERMINAL ERROR!                                                  "
  STOP
END IF

20 FORMAT(2X,"h_R_i = ",F7.5,"m (h_R_i < 3dz)")
21 FORMAT(2X,"Increase h_R_i up to ",F7.5,"m")
IF (h_R_i.LT.6.0D0*dz) THEN
  PRINT 20,  h_R_i
  PRINT 21,  6.0D0*dz
  PRINT *, " TERMINAL ERROR! "
  STOP
END IF

CALL WARNING_FOR_WRITING(NY2)
! ------------------------------------------------------------------------------- !
MAX_NUMBER_OF_ITERATIONS = IDNINT(ZERO_TIME/dt_Plastic)

PRINT *, "      ----------------------------------------  "
PRINT "( '         MAX_NUMBER_OF_ITERATIONS: ', I10 )        ",&
                   MAX_NUMBER_OF_ITERATIONS
PRINT *, "      ----------------------------------------  "
PRINT *, "                                                "
```

```
PRINT *, "              ----------------------------                  "
WRITE(*,"(A)",ADVANCE="NO") "               PRESS 'ENTER' TO CONTINUE"
PRINT *, " "
READ (*,"(A)") IGNORED_INPUT
PRINT *, " "
! ------------------------------------------------------------------------------- !
! Creating log file and making the first entry:
OPEN(unit=8,file="log.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: log.dat! "
  STOP
ELSE
  WRITE (unit=8,fmt=*) 0,0.0D0
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------- !
ALLOCATE(MX1(NX2-2,NX2-2),MX2(NX-2,NX-2),MY1(NY1_corner-2,NY1_corner-2),&
         MY2(NY2-2,NY2-2),VELOCITY_k(NX2,NY2),VELOCITY_kp12(NX2,NY2),&
         VELOCITY_kp12_new(NX2,NY2),VELOCITY_kp1(NX2,NY2),&
         VELOCITY_kp1_new(NX2,NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 1 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(K1(NX2-2),K2(NX-2),L1(NY1_corner-2),L2(NY2-2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 2 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(v1r_ijp1k(NX2),           v1r_ijk(NX2),           v1r_ijm1k(NX2),&
         v1r_ijp1kp12(NX2),        v1r_ijkp12(NX2),        v1r_ijm1kp12(NX2),&
         v1r_c_ijp1k(NX2),         v1r_c_ijk(NX2),         v1r_c_ijm1k(NX2),&
         v1r_c_ijp1kp12(NX2),      v1r_c_ijkp12(NX2),      v1r_c_ijm1kp12(NX2),&
         v2r_ijp1k(NX),            v2r_ijk(NX),            v2r_ijm1k(NX),&
         v2r_ijp1kp12(NX),         v2r_ijkp12(NX),         v2r_ijm1kp12(NX),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 3 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(v1r_ijkp12_new(NX2-2),v1r_c_ijkp12_new(NX2-2),v2r_ijkp12_new(NX-2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 4 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(v1z_ip1jkp12(NY1_corner), v1z_ijkp12(NY1_corner), v1z_im1jkp12(NY1_corner),&
         v1z_ip1jkp1(NY1_corner),  v1z_ijkp1(NY1_corner),  v1z_im1jkp1(NY1_corner),&
         v1z_c_ip1jkp12(NY2),      v1z_c_ijkp12(NY2),      v1z_c_im1jkp12(NY2),&
         v1z_c_ip1jkp1(NY2),       v1z_c_ijkp1(NY2),       v1z_c_im1jkp1(NY2),&
         v2z_ip1jkp12(NY2),        v2z_ijkp12(NY2),        v2z_im1jkp12(NY2),&
         v2z_ip1jkp1(NY2),         v2z_ijkp1(NY2),         v2z_im1jkp1(NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 5 in main and execution terminated!           "
  STOP
END IF

ALLOCATE(v1z_ijkp1_new(NY1_corner-2),v1z_c_ijkp1_new(NY2-2),&
         v2z_ijkp1_new(NY2-2),stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not allocate space! "
  PRINT *," Error code 6 in main and execution terminated!           "
  STOP
END IF
```

```
! ------------------------------------------------------------------------------ !
! Initialization:
! ============================================================================== !
MX1               = 0.0D0
MX2               = 0.0D0
K1                = 0.0D0
K2                = 0.0D0

VELOCITY_k        = 0.0D0
VELOCITY_kp12     = 0.0D0
VELOCITY_kp12_new = 0.0D0
! ------------------------------------------------------------------------------ !
v1r_ijkp12_new    = 0.0D0
v1r_c_ijkp12_new  = 0.0D0
v2r_ijkp12_new    = 0.0D0

! v1r is used in K1 and MX1 -> K1=K1(v1r) and MX1=MX1(v1r) to solve the system
! MX1*v1r_new=K1. "v1r" could be called "v1r_old" since it is the velocity
! from the previous iteration.
v1r_ijp1k         = 0.0D0  ! v1r -> K1 & MX1*v1r_new=K1
v1r_ijk           = 0.0D0
v1r_ijm1k         = 0.0D0
v1r_ijp1kp12      = 0.0D0
v1r_ijkp12        = 0.0D0
v1r_ijm1kp12      = 0.0D0
! ------------------------------------------------------------------------------ !
v1r_c_ijp1k       = 0.0D0  ! v1r_c -> K1(1:x) & MX1(1:x)*v1r_c_new(1:x)=K1(1:x)
v1r_c_ijk         = 0.0D0  ! ..._c -> ..._corner
v1r_c_ijm1k       = 0.0D0
v1r_c_ijp1kp12    = 0.0D0
v1r_c_ijkp12      = 0.0D0
v1r_c_ijm1kp12    = 0.0D0

v2r_ijp1k         = 0.0D0  ! v2r -> K2 & MX2*v2r_new=K2
v2r_ijk           = 0.0D0
v2r_ijm1k         = 0.0D0
v2r_ijp1kp12      = 0.0D0
v2r_ijkp12        = 0.0D0
v2r_ijm1kp12      = 0.0D0
! ============================================================================== !
MY1               = 0.0D0
MY2               = 0.0D0
L1                = 0.0D0
L2                = 0.0D0

VELOCITY_kp1      = 0.0D0
VELOCITY_kp1_new  = 0.0D0
! ------------------------------------------------------------------------------ !
v1z_ijkp1_new     = 0.0D0
v1z_c_ijkp1_new   = 0.0D0
v2z_ijkp1_new     = 0.0D0

! v1z is used in L1 and MY1 -> L1=L1(v1z) and MY1=MY1(v1z) to solve the system
! MY1*v1z_new=L1. "v1z" could be called "v1z_old" since it is the velocity
! from the previous iteration.
v1z_ip1jkp12      = 0.0D0
v1z_ijkp12        = 0.0D0
v1z_im1jkp12      = 0.0D0
v1z_ip1jkp1       = 0.0D0
v1z_ijkp1         = 0.0D0
v1z_im1jkp1       = 0.0D0
! ------------------------------------------------------------------------------ !
v1z_c_ip1jkp12    = 0.0D0  ! v1z_c -> L1(1:y) & MY1(1:y)*v1z_c_new(1:y)=L1(1:y)
v1z_c_ijkp12      = 0.0D0  ! ..._c -> ..._corner
v1z_c_im1jkp12    = 0.0D0
v1z_c_ip1jkp1     = 0.0D0
v1z_c_ijkp1       = 0.0D0
v1z_c_im1jkp1     = 0.0D0

v2z_ip1jkp12      = 0.0D0  ! v2z -> L2 & MY2*v2z_new=L2
v2z_ijkp12        = 0.0D0
v2z_im1jkp12      = 0.0D0
v2z_ip1jkp1       = 0.0D0
v2z_ijkp1         = 0.0D0
```

```
v2z_im1jkp1       = 0.0D0
! ================================================================================ !
WARNING_SIGN      = .FALSE.
FALSE_CONVERGENCE = .FALSE.
! -------------------------------------------------------------------------------- !
! Initialization of boundary condition at t = 0.0 sec:
CALL ANGULAR_VELOCITY(0.0D0,dt,omega)
! Dirichlet boundary condition:

DO i = 1,NX2
  VELOCITY_k(i,1) = omega*DBLE(i-1)*dr
END DO
VELOCITY_k(NX2,:) = R_o*omega

! ################################################################################ !
! In Section 7.11.1 is a detailed description of the algorithm, which is used in   !
! the following.                                                                   !
! -------------------------------------------------------------------------------- !
! Linear approximation to speed up convergence:
DO i = 2,NX1
  a = VELOCITY_k(i,1)
  b = VELOCITY_k(i,NY1_corner)
  DO j = 2,NY1_corner-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(j-1)/DBLE(NY1_corner-1)
  END DO
END DO

DO i = NX1+1,NX1_corner
  NY_corner = y_corner(NX1_corner - i + 1)
  a = VELOCITY_k(i,1)
  b = VELOCITY_k(i,NY_corner)
  DO j = 2,NY_corner-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(j-1)/DBLE(NY_corner-1)
  END DO
END DO

DO j = 2,NY2-1
  a = VELOCITY_k(NX1_corner,j)
  b = VELOCITY_k(NX2,j)
  DO i = NX1_corner+1,NX2-1
    VELOCITY_k(i,j) = a - (a - b)*DBLE(i-NX1_corner)/DBLE(NX2-NX1_corner)
  END DO
END DO

! Neumann boundary condition:
VELOCITY_k(NX1_corner+1:NX2-1,NY2) = (4.0D0 * &
     VELOCITY_k(NX1_corner+1:NX2-1,NY2-1) - &
     VELOCITY_k(NX1_corner+1:NX2-1,NY2-2))/3.0D0

! CHECK OUT IF VELOCITY_k IS OK:
! CALL WRITE2FILE_k(VELOCITY_k,NX2)
! STOP
! -------------------------------------------------------------------------------- !
VELOCITY_kp12     = VELOCITY_k
VELOCITY_kp12_new = VELOCITY_k
VELOCITY_kp1_new  = VELOCITY_k
VELOCITY_kp1      = VELOCITY_k
! ================================================================================ !
! ============================ Begin of CONTINUATION ============================= !
! ================================================================================ !
N_Lambda_MAX = 1
CONTINUATION: DO N_Lambda = 0,N_Lambda_MAX
! Lambda => The Continuation Method (see Section 7.8).
Lambda = DBLE(N_Lambda)/DBLE(N_Lambda_MAX)
PRINT *,"-------------------------------------------------------"
PRINT *,"CONTINUATION:",Lambda

IF (N_Lambda == 0) THEN
  dt  = dt_Newton
  tol = tol_Newton
  tol_RMS_active = tol_Newton
ELSE
  dt  = dt_Plastic
  tol = tol_Plastic
```

```
  tol_RMS_active = tol_RMS
END IF

TIME_INDEPENDENCE = .FALSE.
! Initializing time for each CONTINUATION step:
k = 0
! ============================================================================== !
! ========================== Begin of the time loop ============================ !
! ============================================================================== !
ZERO_TIME_LOOP: DO WHILE (.NOT.TIME_INDEPENDENCE)
CONVERGENCE = .FALSE.
! ------------------------------------------------------
IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT *,"------------------------------------------------------"
  PRINT *,"                                                      "
  PRINT *,"  PSEUDO-TRANSIENT time step: k+1 = ",k+1
  PRINT *,"------------------------------------------------------"
END IF
! ------------------------------------------------------------------------------ !
count = 0
! ============================================================================== !
! ====================== BEGIN OF SUCCESSIVE SUBSTITUTION ====================== !
! ============================================================================== !
! The iteration loop here is because of the non-linearity of the governing
! Equations 7.22 and 7.23. To come around this problem, the successive substitution
! approach is used (see Section 7.8).
SUBSTITUTION: DO WHILE (.NOT.CONVERGENCE)

! If convergence is a problem, then this might help:
! VELOCITY_kp12 = (VELOCITY_k + VELOCITY_kp1_new)/2
count = count + 1
10 FORMAT(4X,"Successive substitution number = ",1(I3,1X))

IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT 10, count
END IF

! ========================== BEGIN OF X-ITERATION ======================== !
! Iteration is made along r-direction (i.e. along the i-direction as
! in A(i,j)). It starts at the bottom of the viscometer i=(2:NX2-1) at
! j = 2 and then move upward with increasing j (see Figure 10.29).

! -------------------------- BEGIN OF X-GAP-AREA ------------------------ !
DO j = 2,NY1_corner-1
  v1r_ijp1k    = VELOCITY_k(:,j+1)
  v1r_ijk      = VELOCITY_k(:,j)
  v1r_ijm1k    = VELOCITY_k(:,j-1)
  v1r_ijp1kp12 = VELOCITY_kp12(:,j+1)
  v1r_ijkp12   = VELOCITY_kp12(:,j)
  v1r_ijm1kp12 = VELOCITY_kp12(:,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,0.0D0,dz,NX2,v1r_ijp1k,v1r_ijk,&
                   v1r_ijm1k,v1r_ijp1kp12,v1r_ijkp12,v1r_ijm1kp12,MX1,K1)
  CALL MATRIX_SOLVER(MX1,K1,v1r_ijkp12_new,NX2-2)

  VELOCITY_kp12_new(2:NX2-1,j) = v1r_ijkp12_new
END DO
! -------------------------- END OF X-GAP-AREA -------------------------- !

! -------------------------- BEGIN OF X-CORNER -------------------------- !
DO j = NY1_corner,NY1-1
  NX_corner                      = x_corner(j - NY1_corner + 1)
  R_i_corner                     = NX_corner*dr
  NX2mNX_corner                  = NX2 - NX_corner
  v1r_c_ijp1k(1:NX2mNX_corner)    = VELOCITY_k(NX_corner+1:NX2,j+1)
  v1r_c_ijk(1:NX2mNX_corner)      = VELOCITY_k(NX_corner+1:NX2,j)
  v1r_c_ijm1k(1:NX2mNX_corner)    = VELOCITY_k(NX_corner+1:NX2,j-1)
  v1r_c_ijp1kp12(1:NX2mNX_corner) = VELOCITY_kp12(NX_corner+1:NX2,j+1)
  v1r_c_ijkp12(1:NX2mNX_corner)   = VELOCITY_kp12(NX_corner+1:NX2,j)
  v1r_c_ijm1kp12(1:NX2mNX_corner) = VELOCITY_kp12(NX_corner+1:NX2,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i_corner,dz,NX2mNX_corner,v1r_c_ijp1k,&
           v1r_c_ijk,v1r_c_ijm1k,v1r_c_ijp1kp12,v1r_c_ijkp12,v1r_c_ijm1kp12,MX1,K1)
  CALL MATRIX_SOLVER(MX1,K1,v1r_c_ijkp12_new,NX2mNX_corner-2)
```

```
  VELOCITY_kp12_new(NX_corner+2:NX2-1,j) = v1r_c_ijkp12_new(1:NX2mNX_corner-2)
END DO
! -------------------------- END OF X-CORNER ---------------------------- !

! -------------------------- BEGIN OF X-OPEN-AREA ----------------------- !
DO j = NY1,NY2-1
  v2r_ijp1k    = VELOCITY_k(NX1_corner:NX2,j+1)
  v2r_ijk      = VELOCITY_k(NX1_corner:NX2,j)
  v2r_ijm1k    = VELOCITY_k(NX1_corner:NX2,j-1)
  v2r_ijp1kp12 = VELOCITY_kp12(NX1_corner:NX2,j+1)
  v2r_ijkp12   = VELOCITY_kp12(NX1_corner:NX2,j)
  v2r_ijm1kp12 = VELOCITY_kp12(NX1_corner:NX2,j-1)

  CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,v2r_ijp1k,v2r_ijk,&
                v2r_ijm1k,v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12,MX2,K2)
  CALL MATRIX_SOLVER(MX2,K2,v2r_ijkp12_new,NX-2)

  VELOCITY_kp12_new(NX1_corner+1:NX2-1,j) = v2r_ijkp12_new
END DO

j = NY2
v2r_ijp1k    = VELOCITY_k(NX1_corner:NX2,j-1)    ! => v(i,j+1) = v(i,j-1)
v2r_ijk      = VELOCITY_k(NX1_corner:NX2,j)
v2r_ijm1k    = VELOCITY_k(NX1_corner:NX2,j-1)    ! => v(i,j+1) = v(i,j-1)
v2r_ijp1kp12 = VELOCITY_kp12(NX1_corner:NX2,j-1) ! => v(i,j+1) = v(i,j-1)
v2r_ijkp12   = VELOCITY_kp12(NX1_corner:NX2,j)
v2r_ijm1kp12 = VELOCITY_kp12(NX1_corner:NX2,j-1) ! => v(i,j+1) = v(i,j-1)

CALL MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,v2r_ijp1k,v2r_ijk,&
               v2r_ijm1k,v2r_ijp1kp12,v2r_ijkp12,v2r_ijm1kp12,MX2,K2)
CALL MATRIX_SOLVER(MX2,K2,v2r_ijkp12_new,NX-2)

VELOCITY_kp12_new(NX1_corner+1:NX2-1,j) = v2r_ijkp12_new
! -------------------------- END OF X-OPEN-AREA ------------------------- !

! --------- PAUSE FOR DEBUGGING ---------
! CALL WRITE2FILE_k(VELOCITY_kp12_new,NX2)
! WRITE(*,"(A)",ADVANCE="NO") " PRESS 'ENTER' TO CONTINUE "
! PRINT *, " "
! READ (*,"(A)") IGNORED_INPUT
! PRINT *, " "

! ========================== END OF X-ITERATION ======================== !

! Updating ..._kp12:
VELOCITY_kp12 = VELOCITY_kp12_new

! ========================== BEGIN OF Y-ITERATION ====================== !
! Iteration is made along z-direction (i.e. along the j-direction as
! in A(i,j)). It starts at the right side of the viscometer j=(2:NY2-1) at
! i = NX2-1 and then moves to the left with decreasing i (see Figure 10.29).

! -------------------------- BEGIN OF Y-OPEN-AREA ----------------------- !
DO i = NX2-1,NX1_corner+1,-1
  v2z_ip1jkp12 = VELOCITY_kp12(i+1,:)
  v2z_ijkp12   = VELOCITY_kp12(i,:)
  v2z_im1jkp12 = VELOCITY_kp12(i-1,:)
  v2z_ip1jkp1  = VELOCITY_kp1(i+1,:)
  v2z_ijkp1    = VELOCITY_kp1(i,:)
  v2z_im1jkp1  = VELOCITY_kp1(i-1,:)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY2,v2z_ip1jkp12,v2z_ijkp12,&
               v2z_im1jkp12,v2z_ip1jkp1,v2z_ijkp1,v2z_im1jkp1,MY2,L2,.TRUE.)
  CALL MATRIX_SOLVER(MY2,L2,v2z_ijkp1_new,NY2-2)

  VELOCITY_kp1_new(i,2:NY2-1) = v2z_ijkp1_new
END DO

! v(i,j+1) = v(i,j-1) =>
VELOCITY_kp1_new(NX1_corner+1:NX2-1,NY2) = VELOCITY_kp1_new(NX1_corner+1:NX2-1,NY2-2)
! -------------------------- END OF Y-OPEN-AREA ------------------------- !

! -------------------------- BEGIN OF Y-CORNER -------------------------- !
```

```
DO i = NX1_corner,NX1+1,-1
  NY_corner                   = y_corner(NX1_corner - i + 1)
  v1z_c_ip1jkp12(1:NY_corner) = VELOCITY_kp12(i+1,1:NY_corner)
  v1z_c_ijkp12(1:NY_corner)   = VELOCITY_kp12(i,1:NY_corner)
  v1z_c_im1jkp12(1:NY_corner) = VELOCITY_kp12(i-1,1:NY_corner)
  v1z_c_ip1jkp1(1:NY_corner)  = VELOCITY_kp1(i+1,1:NY_corner)
  v1z_c_ijkp1(1:NY_corner)    = VELOCITY_kp1(i,1:NY_corner)
  v1z_c_im1jkp1(1:NY_corner)  = VELOCITY_kp1(i-1,1:NY_corner)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY_corner,v1z_c_ip1jkp12,v1z_c_ijkp12,&
               v1z_c_im1jkp12,v1z_c_ip1jkp1,v1z_c_ijkp1,v1z_c_im1jkp1,MY2,L2,.FALSE.)
  CALL MATRIX_SOLVER(MY2,L2,v1z_c_ijkp1_new,NY_corner-2)

  VELOCITY_kp1_new(i,2:NY_corner-1) = v1z_c_ijkp1_new(1:NY_corner-2)
END DO
! -------------------------- END OF Y-CORNER ---------------------------- !

! -------------------------- BEGIN OF Y-GAP-AREA ------------------------ !
DO i = NX1,2,-1
  v1z_ip1jkp12 = VELOCITY_kp12(i+1,1:NY1_corner)
  v1z_ijkp12   = VELOCITY_kp12(i,1:NY1_corner)
  v1z_im1jkp12 = VELOCITY_kp12(i-1,1:NY1_corner)
  v1z_ip1jkp1  = VELOCITY_kp1(i+1,1:NY1_corner)
  v1z_ijkp1    = VELOCITY_kp1(i,1:NY1_corner)
  v1z_im1jkp1  = VELOCITY_kp1(i-1,1:NY1_corner)

  CALL MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY1_corner,v1z_ip1jkp12,v1z_ijkp12,&
                   v1z_im1jkp12,v1z_ip1jkp1,v1z_ijkp1,v1z_im1jkp1,MY1,L1,.FALSE.)
  CALL MATRIX_SOLVER(MY1,L1,v1z_ijkp1_new,NY1_corner-2)

  VELOCITY_kp1_new(i,2:NY1_corner-1) = v1z_ijkp1_new
END DO
! -------------------------- END OF Y-GAP-AREA -------------------------- !

! --------- PAUSE FOR DEBUGGING ---------
! CALL WRITE2FILE_k(VELOCITY_kp12_new,NX2)
! WRITE(*,"(A)",ADVANCE="NO") " PRESS 'ENTER' TO CONTINUE "
! PRINT *, " "
! READ (*,"(A)") IGNORED_INPUT
! PRINT *, " "

! ========================== END OF Y-ITERATION ======================== !

CONVERGENCE = .TRUE.

! Settings for testing of convergence:
RMS      = 0.0D0
vel_norm = 1.0D0

S1: DO j = 2,NY1_corner-1
      DO i = 2,NX1_corner
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S1
        END IF
      END DO
    END DO S1

IF (CONVERGENCE) THEN
S2: DO j = NY1_corner,NY1-1
      NX_corner = x_corner(j - NY1_corner + 1)
      DO i = NX_corner+2,NX1_corner
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S2
        END IF
      END DO
    END DO S2
END IF
```

```
IF (CONVERGENCE) THEN
S3: DO j = NY1_corner,NY2-1
      DO i = NX1_corner+1,NX2-1
        vel_norm =  (VELOCITY_kp1_new(i,j) + VELOCITY_kp1(i,j))/2.0D0 + EPS
        RMS      = ((VELOCITY_kp1_new(i,j) - VELOCITY_kp1(i,j))/vel_norm)**2.0D0
        IF (RMS > tol) THEN
          CONVERGENCE = .FALSE.
          EXIT S3
        END IF
      END DO
    END DO S3
END IF

IF (count == count_max) THEN
  CONVERGENCE       = .TRUE.
  FALSE_CONVERGENCE = .TRUE.
  PRINT *, "WARNING: FALSE CONVERGENCE! TIME STEP k= ", k
  PRINT *, "Maximum amount of successive substitutions is = ", count_max
  PRINT *, " ------------------------------------------------------------ "
  PRINT *, " RECOMMENDATION: Kill this application and reduce the time  "
  PRINT *, " step by an order of magnitude: dt -> dt/10                 "
  PRINT *, " ------------------------------------------------------------ "
END IF

! Updating ..._kp1
VELOCITY_kp1 = VELOCITY_kp1_new

! CALL WRITE2FILE_k(VELOCITY_k,NX2)

END DO SUBSTITUTION
! ============================================================================== !
! ======================== END OF SUCCESSIVE SUBSTITUTION ======================== !
! ============================================================================== !
count_rms = 0
RMS       = 0.0D0
vel_norm  = 1.0D0

DO j = 2,NY1_corner-1
  DO i = 2,NX1_corner
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
  END DO
END DO

DO j = NY1_corner,NY1-1
  NX_corner = x_corner(j - NY1_corner + 1)
  DO i = NX_corner+2,NX1_corner
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
  END DO
END DO

DO j = NY1_corner,NY2-1
  DO i = NX1_corner+1,NX2-1
    count_rms = count_rms + 1
    vel_norm  =  (VELOCITY_kp1(i,j) + VELOCITY_k(i,j))/2.0D0 + EPS
    RMS       = ((VELOCITY_kp1(i,j) - VELOCITY_k(i,j))/vel_norm)**2.0D0 + RMS
  END DO
END DO

RMS = DSQRT(RMS/count_rms)
! ------------
IF (ABS(MOD((k+1),k_OUTPUT_rms)).LT.small_zero) THEN
  PRINT *, "   RMS   =", RMS
  CALL WRITE2FILE_rms(k+1,RMS)
END IF
! ------------
IF (RMS.LT.tol_RMS_active) THEN
  TIME_INDEPENDENCE = .TRUE.
ELSE
  TIME_INDEPENDENCE = .FALSE.
END IF
```

```
! ------------
IF (k == MAX_NUMBER_OF_ITERATIONS) THEN
  TIME_INDEPENDENCE = .TRUE.
  WARNING_SIGN      = .TRUE.
END IF
! ------------
VELOCITY_k = VELOCITY_kp1_new
k = k + 1
! ------------
! CALL WRITE2FILE_k(VELOCITY_k,NX2)
! STOP
END DO ZERO_TIME_LOOP
! ============================================================================== !
! ============================ End of the time loop ============================ !
! ============================================================================== !
END DO CONTINUATION
! ============================================================================== !
! ============================ End of CONTINUATION ============================= !
! ============================================================================== !
IF (WARNING_SIGN) THEN
  PRINT *, " ---------------------------------------------------------- "
  PRINT *, " WARNING: k = MAX_NUMBER_OF_ITERATIONS; See log.dat        "
  PRINT *, " RECOMMENDATION: Rerun this application with reduced time  "
  PRINT *, " step. Try order of magnitude less: dt -> dt/10            "
  PRINT *, " ---------------------------------------------------------- "
END IF

IF (FALSE_CONVERGENCE) THEN
  PRINT *, " ---------------------------------------------------------- "
  PRINT *, " WARNING: FALSE CONVERGENCE WAS ACHIEVED. Reduce the time  "
  PRINT *, " step by order of magnitude: dt -> dt/10 and then rerun    "
  PRINT *, " the application.                                          "
  PRINT *, " ---------------------------------------------------------- "
END IF

PRINT *, " PSEUDO-TRANSIENT CALCULATION FINISHED! "
PRINT *, " ------------------------------------ "

PRINT *, " Number of grid points:", count_rms
PRINT *, "                       "
PRINT *, " WRITING INFORMATION TO FILE, STAND BY... "
CALL WRITE2FILE_k(VELOCITY_k,NX2)
CALL WRITE2FILE_kp1(VELOCITY_k,NX1,NX1_corner,NX2,NY1,NY1_corner,NY2,dr,dz)
PRINT *, " ...DONE! "

! ------------------------------------------------------------------------------ !
! CLEARING SOME MAJOR VARIABLES FROM THE RANDOM ACCESS MEMORY:
DEALLOCATE(x_corner,y_corner,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 7 in main and execution terminated!             "
END IF

DEALLOCATE(MX1,MX2,MY1,MY2,VELOCITY_k,VELOCITY_kp12,VELOCITY_kp12_new,&
           VELOCITY_kp1,VELOCITY_kp1_new,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 8 in main!                                      "
END IF

DEALLOCATE(K1,K2,L1,L2,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 9 in main!                                      "
END IF

DEALLOCATE(v1r_ijp1k,      v1r_ijk,      v1r_ijm1k,&
           v1r_ijp1kp12,   v1r_ijkp12,   v1r_ijm1kp12,&
           v1r_c_ijp1k,    v1r_c_ijk,    v1r_c_ijm1k,&
           v1r_c_ijp1kp12, v1r_c_ijkp12, v1r_c_ijm1kp12,&
           v2r_ijp1k,      v2r_ijk,      v2r_ijm1k,&
           v2r_ijp1kp12,   v2r_ijkp12,   v2r_ijm1kp12,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
```

```
  PRINT *," Error code 10 in main!                                    "
END IF

DEALLOCATE(v1r_ijkp12_new,v1r_c_ijkp12_new,v2r_ijkp12_new,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 11 in main!                                    "
END IF

DEALLOCATE(v1z_ip1jkp12,   v1z_ijkp12,   v1z_im1jkp12,&
           v1z_ip1jkp1,    v1z_ijkp1,    v1z_im1jkp1,&
           v1z_c_ip1jkp12, v1z_c_ijkp12, v1z_c_im1jkp12,&
           v1z_c_ip1jkp1,  v1z_c_ijkp1,  v1z_c_im1jkp1,&
           v2z_ip1jkp12,   v2z_ijkp12,   v2z_im1jkp12,&
           v2z_ip1jkp1,    v2z_ijkp1,    v2z_im1jkp1,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 12 in main!                                    "
END IF

DEALLOCATE(v1z_ijkp1_new,v1z_c_ijkp1_new,v2z_ijkp1_new,stat=problem)
IF (problem/=0) THEN
  PRINT *," MAIN_ROUTINE says: The program could not deallocate space! "
  PRINT *," Error code 13 in main!                                    "
END IF

PRINT *, " EXECUTION FINISHED! "
! ------------------------------------------------------------------------------ !
END PROGRAM MAIN_ROUTINE
! ------------------------------------------------------------------------------ !
```

A.3.3 shear.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: shear.f90 (MODULE) [is the same to what is shown in Appendix A.2.6] !
! This routine calculates the shear rate SR(i,j) from the computed velocity profile !
! VELOCITY_k(i,j). It is the program update.f90 that makes the request.          !
! See Section 7.5 about the formulas for the shear rate (SR). Note that ROS and SR  !
! means the same thing: ROS = rate of shear = SR = shear rate.                   !
! ------------------------------------------------------------------------------ !
MODULE SHEAR_RATE
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: SR
CONTAINS
! ============================================================================== !
SUBROUTINE SR(rp1,r,rm1,dr,dz,V_ij,V_ip1j,V_im1j,V_ijp1,V_ijm1,V_ip1jp1,V_ip1jm1,&
              V_im1jp1,V_im1jm1,SR_ij,SR_ip12j,SR_im12j,SR_ijp12,SR_ijm12)

DOUBLE PRECISION,INTENT(IN)  ::  rp1,r,rm1,dr,dz,V_ij,&
                                 V_ip1j,    V_im1j,    V_ijp1,    V_ijm1,&
                                 V_ip1jp1,  V_ip1jm1,  V_im1jp1,  V_im1jm1
DOUBLE PRECISION,INTENT(OUT) ::  SR_ij,     SR_ip12j,  SR_im12j,  SR_ijp12, SR_ijm12
DOUBLE PRECISION             ::  SR1_ij,    SR2_ij,    SR1_ip12j, SR2_ip12j,&
                                 SR1_im12j, SR2_im12j, SR1_ijp12,&
                                 SR2_ijp12, SR1_ijm12, SR2_ijm12
! ------------------------------------------------------------------------------ !
SR1_ij    = (V_ip1j - V_im1j)/(2.0D0*dr) - V_ij/r
SR2_ij    = (V_ijp1 - V_ijm1)/(2.0D0*dz)
SR_ij     = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)

SR1_ip12j = (V_ip1j - V_ij)/dr - (V_ip1j + V_ij)/(rp1 + r)
SR2_ip12j = (V_ip1jp1 + V_ijp1 - V_ip1jm1 - V_ijm1)/(4.0D0*dz)
SR_ip12j  = DSQRT(SR1_ip12j**2.0D0 + SR2_ip12j**2.0D0)

SR1_im12j = (V_ij - V_im1j)/dr - (V_ij + V_im1j)/(r + rm1)
SR2_im12j = (V_ijp1 + V_im1jp1 - V_ijm1 - V_im1jm1)/(4.0D0*dz)
SR_im12j  = DSQRT(SR1_im12j**2.0D0 + SR2_im12j**2.0D0)

SR1_ijp12 = (V_ip1jp1 + V_ip1j - V_im1jp1 - &
             V_im1j)/(4.0D0*dr) - (V_ijp1 + V_ij)/(2.0D0*r)
SR2_ijp12 = (V_ijp1 - V_ij)/dz
SR_ijp12  = DSQRT(SR1_ijp12**2.0D0 + SR2_ijp12**2.0D0)

SR1_ijm12 = (V_ip1j + V_ip1jm1 - V_im1j - &
             V_im1jm1)/(4.0D0*dr) - (V_ij + V_ijm1)/(2.0D0*r)
SR2_ijm12 = (V_ij - V_ijm1)/dz
SR_ijm12  = DSQRT(SR1_ijm12**2.0D0 + SR2_ijm12**2.0D0)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE SR
! ============================================================================== !
END MODULE SHEAR_RATE
```

A.3.4 motion.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: motion.f90 (MODULE)                                                 !
! The information about the (constant) angular velocity "omega" is requested by the !
! routine main.f90.                                                              !
! ------------------------------------------------------------------------------ !
MODULE ROTATION
  IMPLICIT NONE
  PRIVATE
  PUBLIC ::  ANGULAR_VELOCITY
CONTAINS
! ============================================================================== !
SUBROUTINE ANGULAR_VELOCITY(double_prec_k,dt,omega)

DOUBLE PRECISION,INTENT(IN)     :: double_prec_k,dt
DOUBLE PRECISION,INTENT(OUT)    :: omega
DOUBLE PRECISION                :: PI,f
! ------------------------------------------------------------------------------ !
! The terms dt and double_prec_k are a legacy from Appendix A.2.
PI    = DACOS(-1.0D0)
f     = 0.5D0
omega = 2*PI*f ! rad/s (for example, omega = 3.0D0 rad/s)
! ------------------------------------------------------------------------------ !
END SUBROUTINE ANGULAR_VELOCITY
! ============================================================================== !
END MODULE ROTATION
! ------------------------------------------------------------------------------ !
```

A.3.5 update.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: update.f90 (MODULE)                                                 !
! This file sets up the system of algebraic Equations 7.28 to 7.31. This file also  !
! contains the Thomas algorithm that is used in solving this system.             !
! ------------------------------------------------------------------------------ !
MODULE MATRIX
  USE SHEAR_RATE
  USE SHEAR_VISCOSITY
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: MATRIX_UPDATE_X,MATRIX_UPDATE_Y,MATRIX_SOLVER
CONTAINS
! ============================================================================== !
SUBROUTINE MATRIX_UPDATE_X(rho,k,dt,Lambda,dr,R_i,dz,NX,vr_ijp1k,vr_ijk,&
                      vr_ijm1k,vr_ijp1kp12,vr_ijkp12,vr_ijm1kp12,M,K_M)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(OUT) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)   :: K_M

DOUBLE PRECISION,DIMENSION(:),INTENT(IN)    :: vr_ijp1k,vr_ijk,&
                                               vr_ijm1k,vr_ijp1kp12,&
                                               vr_ijkp12,vr_ijm1kp12

DOUBLE PRECISION,INTENT(IN)                 :: dt,dr,R_i,dz,Lambda,rho
INTEGER,INTENT(IN)                          :: k,NX

DOUBLE PRECISION :: BETA,THETA,CHI,rp1,r,rm1

DOUBLE PRECISION :: V_ijk,          V_ip1jk,        V_im1jk,&
                    V_ijp1k,        V_ijm1k,        V_ip1jp1k,&
                    V_ip1jm1k,      V_im1jp1k,      V_im1jm1k,&
                    V_ijkp12,       V_ip1jkp12,     V_im1jkp12,&
                    V_ijp1kp12,     V_ijm1kp12,     V_ip1jp1kp12,&
                    V_ip1jm1kp12,   V_im1jp1kp12,   V_im1jm1kp12

DOUBLE PRECISION :: SR_ijk,         SR_ip12jk,      SR_im12jk,&
                    SR_ijp12k,      SR_ijm12k,&
                    SR_ijkp12,      SR_ip12jkp12,   SR_im12jkp12,&
                    SR_ijp12kp12,   SR_ijm12kp12

DOUBLE PRECISION :: ETA_ijk,        ETA_ip12jk,     ETA_im12jk,&
                    ETA_ijp12k,     ETA_ijm12k,&
                    ETA_ijkp12,     ETA_ip12jkp12,  ETA_im12jkp12,&
                    ETA_ijp12kp12,  ETA_ijm12kp12

DOUBLE PRECISION :: A_kp12,B_kp12,C_kp12,D_k,E_k,F_k,KK
DOUBLE PRECISION :: TIME_k,TIME_kp12
INTEGER          :: i

BETA      = dt/(2.0D0*dr*rho)
```

```
CHI       = dt/(2.0D0*dz*rho)

TIME_k    =  DBLE(k)*dt
TIME_kp12 = (DBLE(k) + 0.5D0)*dt

! ------------------------------------------------------------------------------ !
! The following resetting is very important in order to avoid programming        !
! error when modifying the code. Incorrect programming will most likely lead     !
! to an additional zeros being incorporated into M and K_M which in turn would   !
! lead to singularity problems that would then be reported by MATRIX_SOLVER.     !
! M     =  0.0D0 ! Resetting matrix!                                             !
! K_M   =  0.0D0 ! Resetting vector!                                             !
! ------------------------------------------------------------------------------ !

! ------------------------------------------------------------------------------ !
! i = 1    => i = 2 in main.f90, i.e. near the center of the viscometer.
! i = NX-2 => j = NX2-1 in main.f90, i.e. near the outer cylinder.
DO i = 1,NX-2

  rp1   =  DBLE(i+1)*dr + R_i
  r     =  DBLE(i)*dr + R_i
  rm1   =  DBLE(i-1)*dr + R_i
  THETA =  dt/(r*rho)

  V_im1jk      = vr_ijk(i)
  V_ijk        = vr_ijk(i+1)
  V_ip1jk      = vr_ijk(i+2)
  V_im1jp1k    = vr_ijp1k(i)
  V_ijp1k      = vr_ijp1k(i+1)
  V_ip1jp1k    = vr_ijp1k(i+2)
  V_im1jm1k    = vr_ijm1k(i)
  V_ijm1k      = vr_ijm1k(i+1)
  V_ip1jm1k    = vr_ijm1k(i+2)

  V_im1jkp12   = vr_ijkp12(i)
  V_ijkp12     = vr_ijkp12(i+1)
  V_ip1jkp12   = vr_ijkp12(i+2)
  V_im1jp1kp12 = vr_ijp1kp12(i)
  V_ijp1kp12   = vr_ijp1kp12(i+1)
  V_ip1jp1kp12 = vr_ijp1kp12(i+2)
  V_im1jm1kp12 = vr_ijm1kp12(i)
  V_ijm1kp12   = vr_ijm1kp12(i+1)
  V_ip1jm1kp12 = vr_ijm1kp12(i+2)

  CALL SR(rp1,r,rm1,dr,dz,V_ijk,V_ip1jk,V_im1jk,V_ijp1k,V_ijm1k,&
          V_ip1jp1k,V_ip1jm1k,V_im1jp1k,V_im1jm1k,&
          SR_ijk,SR_ip12jk,SR_im12jk,SR_ijp12k,SR_ijm12k)

  CALL SR(rp1,r,rm1,dr,dz,V_ijkp12,V_ip1jkp12,V_im1jkp12,V_ijp1kp12,V_ijm1kp12,&
          V_ip1jp1kp12,V_ip1jm1kp12,V_im1jp1kp12,V_im1jm1kp12,&
          SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,SR_ijp12kp12,SR_ijm12kp12)

  CALL ETA(TIME_k,Lambda,SR_ijk,SR_ip12jk,SR_im12jk,SR_ijp12k,SR_ijm12k,&
           ETA_ijk,ETA_ip12jk,ETA_im12jk,ETA_ijp12k,ETA_ijm12k)

  CALL ETA(TIME_kp12,Lambda,SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,&
           SR_ijp12kp12,SR_ijm12kp12,ETA_ijkp12,ETA_ip12jkp12,&
           ETA_im12jkp12,ETA_ijp12kp12,ETA_ijm12kp12)

  ! Equations 7.32 to 7.34:
  A_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr - 1.0D0/(rp1+r)) + (THETA*ETA_ijkp12)/(2.0D0*dr)
  B_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr + 1.0D0/(rp1+r)) + &
           (BETA*ETA_im12jkp12)*(1.0D0/dr - 1.0D0/(r+rm1)) + (THETA*ETA_ijkp12)/r
  C_kp12 = (BETA*ETA_im12jkp12)*(1.0D0/dr + 1.0D0/(r+rm1)) - (THETA*ETA_ijkp12)/(2.0D0*dr)

  ! Equations 7.35 to 7.37:
  D_k    = (CHI*ETA_ijp12k)/dz
  E_k    = (CHI*ETA_ijp12k)/dz + (CHI*ETA_ijm12k)/dz
  F_k    = (CHI*ETA_ijm12k)/dz

  ! Equation 7.29:
  KK = - D_k*V_ijp1k - (1.0D0 - E_k)*V_ijk - F_k*V_ijm1k

  ! Equation 7.28:
```

```
  IF (i == 1) THEN
    M(i,i)   = -(1.0D0 + B_kp12)
    M(i,i+1) = A_kp12
    K_M(i)   = KK - C_kp12*V_im1jkp12
  ELSE IF (i == NX-2) THEN
    M(i,i-1) = C_kp12
    M(i,i)   = -(1.0D0 + B_kp12)
    K_M(i)   = KK - A_kp12*V_ip1jkp12
  ELSE
    M(i,i-1) = C_kp12
    M(i,i)   = -(1.0D0 + B_kp12)
    M(i,i+1) = A_kp12
    K_M(i)   = KK
  END IF

END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE MATRIX_UPDATE_X
! ============================================================================== !
SUBROUTINE MATRIX_UPDATE_Y(rho,k,dt,Lambda,dr,i,dz,NY,vz_ip1jkp12,vz_ijkp12,&
                     vz_im1jkp12,vz_ip1jkp1,vz_ijkp1,vz_im1jkp1,M,L,Neumann)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(OUT) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)   :: L

DOUBLE PRECISION,DIMENSION(:),INTENT(IN)    :: vz_im1jkp12,vz_ijkp12,&
                                               vz_ip1jkp12,vz_im1jkp1,&
                                               vz_ijkp1,vz_ip1jkp1

DOUBLE PRECISION,INTENT(IN)                 :: dt,dr,dz,Lambda,rho
INTEGER,INTENT(IN)                          :: k,NY,i

LOGICAL,INTENT(IN)                          :: Neumann

DOUBLE PRECISION :: BETA,THETA,CHI,rp1,r,rm1

DOUBLE PRECISION :: V_ijkp12,       V_ip1jkp12,     V_im1jkp12,&
                    V_ijp1kp12,     V_ijm1kp12,     V_ip1jp1kp12,&
                    V_ip1jm1kp12,   V_im1jp1kp12,   V_im1jm1kp12,&
                    V_ijkp1,        V_ip1jkp1,      V_im1jkp1,&
                    V_ijp1kp1,      V_ijm1kp1,      V_ip1jp1kp1,&
                    V_ip1jm1kp1,    V_im1jp1kp1,    V_im1jm1kp1

DOUBLE PRECISION :: SR_ijkp12,      SR_ip12jkp12,   SR_im12jkp12,&
                    SR_ijp12kp12,   SR_ijm12kp12,&
                    SR_ijkp1,       SR_ip12jkp1,    SR_im12jkp1,&
                    SR_ijp12kp1,    SR_ijm12kp1

DOUBLE PRECISION :: ETA_ijkp12,     ETA_ip12jkp12,  ETA_im12jkp12,&
                    ETA_ijp12kp12,  ETA_ijm12kp12,&
                    ETA_ijkp1,      ETA_ip12jkp1,   ETA_im12jkp1,&
                    ETA_ijp12kp1,   ETA_ijm12kp1

DOUBLE PRECISION :: A_kp12,B_kp12,C_kp12,D_kp1,E_kp1,F_kp1,LL
DOUBLE PRECISION :: TIME_kp12,TIME_kp1
INTEGER          :: j

BETA      = dt/(2.0D0*dr*rho)
CHI       = dt/(2.0D0*dz*rho)
TIME_kp12 = (DBLE(k) + 0.5D0)*dt
TIME_kp1  = (DBLE(k) + 1.0D0)*dt

! ------------------------------------------------------------------------------ !
! The following resetting is very important in order to avoid programming        !
! error when modifying the code. Incorrect programming will most likely lead     !
! to an additional zeros being incorporated into L and M which in turn would     !
! lead to singularity problems that would then be reported by MATRIX_SOLVER.     !
! M     =  0.0D0 ! Resetting matrix!                                             !
! L     =  0.0D0 ! Resetting vector!                                             !
! ------------------------------------------------------------------------------ !

rp1   =  DBLE(i)*dr
r     =  DBLE(i-1)*dr
```

```
rm1   =  DBLE(i-2)*dr
THETA =  dt/(r*rho)

! ------------------------------------------------------------------------------ !
! j = 1    => j = 2 in main.f90, i.e. near the bottom plate of the viscometer.
! j = NY-2 => j = NY2-1 in main.f90, i.e. near the top of the viscometer.
DO j = 1,NY-2

  V_im1jkp12   = vz_im1jkp12(j+1)
  V_ijkp12     = vz_ijkp12(j+1)
  V_ip1jkp12   = vz_ip1jkp12(j+1)
  V_im1jp1kp12 = vz_im1jkp12(j+2)
  V_ijp1kp12   = vz_ijkp12(j+2)
  V_ip1jp1kp12 = vz_ip1jkp12(j+2)
  V_im1jm1kp12 = vz_im1jkp12(j)
  V_ijm1kp12   = vz_ijkp12(j)
  V_ip1jm1kp12 = vz_ip1jkp12(j)

  V_im1jkp1    = vz_im1jkp1(j+1)
  V_ijkp1      = vz_ijkp1(j+1)
  V_ip1jkp1    = vz_ip1jkp1(j+1)
  V_im1jp1kp1  = vz_im1jkp1(j+2)
  V_ijp1kp1    = vz_ijkp1(j+2)
  V_ip1jp1kp1  = vz_ip1jkp1(j+2)
  V_im1jm1kp1  = vz_im1jkp1(j)
  V_ijm1kp1    = vz_ijkp1(j)
  V_ip1jm1kp1  = vz_ip1jkp1(j)

  CALL SR(rp1,r,rm1,dr,dz,V_ijkp12,V_ip1jkp12,V_im1jkp12,V_ijp1kp12,V_ijm1kp12,&
          V_ip1jp1kp12,V_ip1jm1kp12,V_im1jp1kp12,V_im1jm1kp12,&
          SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,SR_ijp12kp12,SR_ijm12kp12)

  CALL SR(rp1,r,rm1,dr,dz,V_ijkp1,V_ip1jkp1,V_im1jkp1,V_ijp1kp1,V_ijm1kp1,&
          V_ip1jp1kp1,V_ip1jm1kp1,V_im1jp1kp1,V_im1jm1kp1,&
          SR_ijkp1,SR_ip12jkp1,SR_im12jkp1,SR_ijp12kp1,SR_ijm12kp1)

  CALL ETA(TIME_kp12,Lambda,SR_ijkp12,SR_ip12jkp12,SR_im12jkp12,&
           SR_ijp12kp12,SR_ijm12kp12,ETA_ijkp12,ETA_ip12jkp12,&
           ETA_im12jkp12,ETA_ijp12kp12,ETA_ijm12kp12)

  CALL ETA(TIME_kp1,Lambda,SR_ijkp1,SR_ip12jkp1,SR_im12jkp1,SR_ijp12kp1,SR_ijm12kp1,&
           ETA_ijkp1,ETA_ip12jkp1,ETA_im12jkp1,ETA_ijp12kp1,ETA_ijm12kp1)

  ! Equations 7.32 to 7.34:
  A_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr - 1.0D0/(rp1+r)) + (THETA*ETA_ijkp12)/(2.0D0*dr)
  B_kp12 = (BETA*ETA_ip12jkp12)*(1.0D0/dr + 1.0D0/(rp1+r)) + &
           (BETA*ETA_im12jkp12)*(1.0D0/dr - 1.0D0/(r+rm1)) + (THETA*ETA_ijkp12)/r
  C_kp12 = (BETA*ETA_im12jkp12)*(1.0D0/dr + 1.0D0/(r+rm1)) - (THETA*ETA_ijkp12)/(2.0D0*dr)

  ! Equations 7.35 to 7.37:
  D_kp1  = (CHI*ETA_ijp12kp1)/dz
  E_kp1  = (CHI*ETA_ijp12kp1)/dz + (CHI*ETA_ijm12kp1)/dz
  F_kp1  = (CHI*ETA_ijm12kp1)/dz

  ! Equation 7.31:
  LL = - A_kp12*V_ip1jkp12 - (1.0D0 - B_kp12)*V_ijkp12 - C_kp12*V_im1jkp12

  ! Equation 7.30:
  IF (j == 1) THEN
    M(j,j)   = -(1.0D0 + E_kp1)
    M(j,j+1) = D_kp1
    L(j)     = LL - F_kp1*V_ijm1kp1
  ELSE IF ((j.EQ.NY-2).AND.Neumann) THEN ! (Equation 7.46)
    M(j,j-1) = 2.0D0*F_kp1
    M(j,j)   = -(1.0D0 + E_kp1)
    L(j)     = LL
  ELSE IF (j == NY-2) THEN
    M(j,j-1) = F_kp1
    M(j,j)   = -(1.0D0 + E_kp1)
    L(j)     = LL - D_kp1*V_ijp1kp1
  ELSE
    M(j,j-1) = F_kp1
    M(j,j)   = -(1.0D0 + E_kp1)
    M(j,j+1) = D_kp1
```

```
    L(j)     = LL
  END IF

END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE MATRIX_UPDATE_Y
! ============================================================================== !
SUBROUTINE MATRIX_SOLVER(M,D,v,dim)
! Subroutine that solves the trigonal system "M * v = D" with the
! Thomas algorithm (also known as the "Crout reduction for tridiagonal linear
! systems" - algorithm).
! M ->  Left side of the linear system (a tridiagonal array).
! D ->  Right side of the linear system (a vector).
! v ->  The variable to be solved: v = (M)^(-1) * D.

DOUBLE PRECISION,DIMENSION(:,:),INTENT(INOUT) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(INOUT)   :: D
DOUBLE PRECISION,DIMENSION(:),INTENT(OUT)     :: v
INTEGER,INTENT(IN)                            :: dim

DOUBLE PRECISION                              :: coef
INTEGER                                       :: i
! ------------------------------------------------------------------------------ !
v = -1.0D0
! ------------------------------------------------------------------------------ !
! Forward and then back substitution:
DO i = 1,dim-1
  coef       = M(i+1,(i-1)+1)/M(i,i)
  M(i+1,i+1) = M(i+1,i+1) - coef*M(i,i+1)
  D(i+1)     = D(i+1) - coef*D(i)
END DO
v(dim) = D(dim)/M(dim,dim)
DO i = dim-1,1,-1
  v(i) = (D(i) - M(i,i+1)*v(i+1))/M(i,i)
END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE MATRIX_SOLVER
! ============================================================================== !
END MODULE MATRIX
! ------------------------------------------------------------------------------ !
```

A.3.6 write2f.f90

```
! ------------------------------------------------------------------------------ !
!                                                                                !
!     Copyright (C) 2002, Jon E. Wallevik, The Norwegian University of           !
!                         Science and Technology (NTNU).                         !
!                                                                                !
!     This file is part of Viscometric-ViscoPlastic-Flow (VVPF).                 !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is free software; you can redistribute it   !
!     and/or modify it under the terms of the GNU General Public License as      !
!     published by the Free Software Foundation; either version 2 of the         !
!     License, or (at your option) any later version.                            !
!                                                                                !
!     Viscometric-ViscoPlastic-Flow, is distributed in the hope that it will be  !
!     useful, but WITHOUT ANY WARRANTY; without even the implied warranty of     !
!     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General   !
!     Public License for more details.                                           !
!                                                                                !
!     You should have received a copy of the GNU General Public License          !
!     along with Viscometric-ViscoPlastic-Flow; if not, write to the Free        !
!     Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,             !
!     MA  02111-1307  USA                                                        !
!                                                                                !
! ------------------------------------------------------------------------------ !
! File name: write2f.f90 (MODULE)                                                !
! This file takes care of writing all computed data into the different files.    !
! It is only the source main.f90 that makes such request.                       !
! ------------------------------------------------------------------------------ !
MODULE WRITE_INFORMATION
  USE SHEAR_VISCOSITY
  IMPLICIT NONE
  PRIVATE
  PUBLIC :: WARNING_FOR_WRITING,WRITE2FILE_k,WRITE2FILE_kp1,&
            WRITE2FILE_debug,WRITE2FILE_rms
CONTAINS
! ============================================================================== !
SUBROUTINE WARNING_FOR_WRITING(NY)
INTEGER,INTENT(IN) :: NY

IF (NY > 500) THEN
  PRINT *, " ERROR: NY2 > 500 ( NY2 = ",NY,")"
  PRINT *, " FORMAT STATEMENT IN THE FILE 'write2f.f90' IS TO SHORT:     "
  PRINT *, " ERROR -> 10 FORMAT(1X,500(F7.4,1X))                         "
  PRINT *, " PLEASE MAKE THE NECESSARY ADJUSTMENT IN ALL THE SUBROUTINES "
  PRINT *, " OF THIS FILE. TERMINAL ERROR!                               "
  STOP
END IF

RETURN
END SUBROUTINE WARNING_FOR_WRITING
! ============================================================================== !
SUBROUTINE WRITE2FILE_k(VELOCITY,NX)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN) :: VELOCITY
INTEGER,INTENT(IN)                         :: NX
INTEGER                                    :: problem,i
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F7.4,1X))
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vel_testing.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_testing.dat! "
  RETURN
ELSE
  DO i = 1,NX
    WRITE (unit=8,fmt=10) VELOCITY(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
```

```
END SUBROUTINE WRITE2FILE_k
! ========================================================================== !
SUBROUTINE WRITE2FILE_kp1(V,NX1,NX1_corner,NX2,NY1,NY1_corner,NY2,dr,dz)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN)  :: V
INTEGER,INTENT(IN)                          :: NX1,NX1_corner,NX2,&
                                               NY1,NY1_corner,NY2
DOUBLE PRECISION,INTENT(IN)                 :: dr,dz

DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:,:) :: SR,ETA,von_Mises
DOUBLE PRECISION,ALLOCATABLE,DIMENSION(:)   :: TORQUE_ON_DISK

DOUBLE PRECISION  :: ETA_tmp,SR_tmp,TIME,Lambda,r,PI,Measured_Torque,&
                     shear_stress_theta_r
INTEGER           :: problem,i,j,k
! -------------------------------------------------------------------------- !
PI = DACOS(-1.0D0)
! -------------------------------------------------------------------------- !
ALLOCATE(SR(NX2,NY2),ETA(NX2,NY2),von_Mises(NX2,NY2),stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_kp1: The program could not allocate space! "
  PRINT *," Error code 1 in write2f and execution terminated!     "
  STOP
END IF

ALLOCATE(TORQUE_ON_DISK(NX1),stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_kp1: The program could not allocate space! "
  PRINT *," Error code 2 in write2f and execution terminated!     "
  STOP
END IF

SR             = 0.0D0
ETA            = 0.0D0
von_Mises      = 0.0D0
TORQUE_ON_DISK = 0.0D0

TIME   = 0.0D0
Lambda = 1.0D0
! -------------------------------------------------------------------------- !
! CALCULATING THE SHEAR RATE AT ALL POINTS: SR
CALL SR_PROFILE(V,NX1,NX1_corner,NX2,NY1,NY1_corner,NY2,dr,dz,SR)
! -------------------------------------------------------------------------- !
! CALCULATING THE SHEAR VISCOSITY AT ALL POINTS:
ETA_tmp = 0.0D0
DO i=1,NX2
  DO j=1,NY2
    SR_tmp   = SR(i,j)
    CALL VISCOSITY(TIME,Lambda,SR_tmp,ETA_tmp)
    ETA(i,j) = ETA_tmp
  END DO
END DO
! -------------------------------------------------------------------------- !
! CALCULATING THE von Mises SHEAR STRESS AT ALL POINTS:
DO i=1,NX2
  DO j=1,NY2
    von_Mises(i,j) = SR(i,j)*ETA(i,j)
  END DO
END DO
! -------------------------------------------------------------------------- !
! CALCULATING THE TORQUE ON THE TOP PLATE (z = h_gap):
j = NY1_corner
DO i=2,NX1-1
  r = DBLE(i-1)*dr
  shear_stress_theta_r = - ETA(i,j)*((-4.0D0*V(i,j-1) + V(i,j-2) &
                         + 3.0D0*V(i,j))/(2.0D0*dz))
  TORQUE_ON_DISK(i)    = r*(shear_stress_theta_r*dr*(2*PI*r))
END DO
! ------------------------------------------------------------
i = NX1
r = DBLE(i-1)*dr
shear_stress_theta_r = - ETA(i,j)*((-4.0D0*V(i,j-1) + V(i,j-2) &
                       + 3.0D0*V(i,j))/(2.0D0*dz))
TORQUE_ON_DISK(NX1)  = r*(shear_stress_theta_r*(dr/2)*(2*PI*r))
```

```
! ------------------------------------------------------------------------------ !
Measured_Torque = 0.0D0
DO i=1,NX1
  Measured_Torque = TORQUE_ON_DISK(i) + Measured_Torque
END DO
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,500(F10.4,1X))
12 FORMAT(1X,500(F14.8,1X))
16 FORMAT(1X,500(F14.4,1X))
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vel_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=10) V(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="SR_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: SR_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=10) SR(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="ETA_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: ETA_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=16) ETA(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="vonMises_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vonMises_t0.dat! "
  RETURN
ELSE
  DO i = 1,NX2
    WRITE (unit=8,fmt=10) von_Mises(i,:)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
PRINT *, " ------------------------------------------------------- "
PRINT *, " Measured torque (on the bottom disk plate):             "
PRINT *,   Measured_Torque, " Nm i_z                               "
PRINT *, " ------------------------------------------------------- "
PRINT *, "                                                         "
! ------------------------------------------------------------------------------ !
24 FORMAT(1X,"4) Measured torque (on the bottom disk plate) = ",(F14.10,1X),"Nm")
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="TORQUE_t0.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: TORQUE_t0.dat! "
```

```
  RETURN
ELSE
  WRITE (unit=8,fmt=24) Measured_Torque
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="TORQUE_disc.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: TORQUE_disc.dat! "
  RETURN
ELSE
  WRITE (unit=8,fmt=12) TORQUE_ON_DISK(1:NX1)
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
DEALLOCATE(SR,ETA,von_Mises,stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_kp1: The program could not deallocate space! "
  PRINT *," Error code 3 in write2f!                                 "
END IF

DEALLOCATE(TORQUE_ON_DISK,stat=problem)
IF (problem/=0) THEN
  PRINT *," WRITE2FILE_kp1: The program could not deallocate space! "
  PRINT *," Error code 4 in write2f!                                 "
END IF
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_kp1
! ============================================================================== !
SUBROUTINE WRITE2FILE_rms(k,rms)

DOUBLE PRECISION,INTENT(IN) :: rms
INTEGER,INTENT(IN)          :: k
INTEGER                     :: problem
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="log.dat",status="old",action="write",&
     position="append",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not write into the existing file: log.dat! "
  RETURN
ELSE
  WRITE (unit=8,fmt=*) k,rms
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_rms
! ============================================================================== !
SUBROUTINE WRITE2FILE_debug(M,K_M,v_new,N)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN) :: M
DOUBLE PRECISION,DIMENSION(:),INTENT(IN)   :: K_M,v_new
INTEGER,INTENT(IN)                         :: N
INTEGER                                    :: problem,i
! ------------------------------------------------------------------------------ !
10 FORMAT(1X,201(F10.4,1X))
11 FORMAT(1X,F10.4)
! ------------------------------------------------------------------------------ !
OPEN(unit=8,file="MM_debug.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: MM_debug.dat! "
  RETURN
ELSE
  DO i = 1,N
    WRITE (unit=8,fmt=10) M(i,:)
  END DO
END IF
```

```
CLOSE (UNIT=8)
! ------------------------------------------------------
OPEN(unit=8,file="KK_debug.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: KK_debug.dat! "
  RETURN
ELSE
  DO i = 1,N
    WRITE (unit=8,fmt=11) K_M(i)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------
OPEN(unit=8,file="vel_debug.dat",status="replace",action="write",&
     position="rewind",iostat=problem)
IF (problem/=0) THEN
  PRINT *," Could not create the file: vel_debug.dat! "
  RETURN
ELSE
  DO i = 1,N
    WRITE (unit=8,fmt=11) v_new(i)
  END DO
END IF

CLOSE (UNIT=8)
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE WRITE2FILE_debug
! ============================================================================== !
SUBROUTINE SR_PROFILE(V,NX1,NX1_corner,NX2,NY1,NY1_corner,NY2,dr,dz,SR)

DOUBLE PRECISION,DIMENSION(:,:),INTENT(IN)  :: V
DOUBLE PRECISION,DIMENSION(:,:),INTENT(OUT) :: SR

INTEGER,INTENT(IN)                          :: NX1,NX1_corner,NX2,&
                                               NY1,NY1_corner,NY2
DOUBLE PRECISION,INTENT(IN)                 :: dr,dz

INTEGER                                     :: i,j,k,problem
DOUBLE PRECISION                            :: r,SR1_ij,SR2_ij,EPS
! ------------------------------------------------------------------------------ !
EPS = 1.0D-15
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE IN THE BULK:
! See Section 7.5 about the formulas for the shear rate (SR). Note that ROS and SR
! means the same thing: ROS = rate of shear = SR = shear rate.
! ------------------------------------------------------------------------------ !
DO i=2,NX2-1
  r = DBLE(i-1)*dr
  DO j=2,NY2-1
    SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
    SR2_ij  = (V(i,j+1) - V(i,j-1))/(2.0D0*dz)
    SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
  END DO
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE ON THE BOTTOM PLATE (z=0):
j = 1
DO i=2,NX2-1
  r = DBLE(i-1)*dr
  SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (4.0D0*V(i,j+1) - V(i,j+2) - 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE "LEFT" WALL (r=R_i):
! SR2_ij is actually zero since the Dirichlet boundary
! condition is not chancing with z!
i = NX1_corner
r = DBLE(i-1)*dr
DO j=NY1+1,NY2-1
  SR1_ij  = (4.0D0*V(i+1,j) - V(i+2,j) - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
```

```
  SR2_ij  = (V(i,j+1) - V(i,j-1))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ---------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE "RIGHT" WALL (r=R_o):
! SR2_ij is actually zero since the Dirichlet boundary
! condition is not chancing with z!
i = NX2
r = DBLE(i-1)*dr
DO j=2,NY2-1
  SR1_ij  = (-4.0D0*V(i-1,j) + V(i-2,j) + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (V(i,j+1) - V(i,j-1))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ---------------------------------------------------------
! CALCULATING THE SHEAR RATE ON TOP PLATE (z=h_gap):
j = NY1_corner
DO i=2,NX1-1
  r = DBLE(i-1)*dr
  SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ---------------------------------------------------------
! CALCULATING THE SHEAR RATE AT THE OPEN BOUNDARY (z=H):
j = NY2
DO i=NX1_corner+1,NX2-1
  r = DBLE(i-1)*dr
  SR1_ij  = (V(i+1,j) - V(i-1,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE AT THE CENTER OF THE BOTTOM PLATE (r=0,z=0):
! This calculation is redundant since the rate of shear at the center line
! ($r=0 \forall z \in [0,H]$) is zero due to symmetry in the r-direction and
! due to the Dirichlet boundary condition $v_{\rm i,j}=0$ at the center line.
! i = 1
! j = 1
! r = DBLE(i-1)*dr
! SR1_ij  = (V(i+1,j) - V(i+1,j))/(2.0D0*dr) - V(i,j)/r ! due to symmetry
! SR2_ij  = ( 4.0D0*V(i,j+1) - V(i,j+2) - 3.0D0*V(i,j))/(2.0D0*dz)
! SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ------------
! Rather enforcing a zero rate of shear at the center line:
SR(1,1:NY2) = 0.0D0
! ------------------------------------------------------------------------------ !
! CALCULATING THE SHEAR RATE ON THE LOWER RIGHT CORNER (r=R_o,z=0):
i = NX2
j = 1
r = DBLE(i-1)*dr
SR1_ij  = (-4.0D0*V(i-1,j) + V(i-2,j) + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = ( 4.0D0*V(i,j+1) - V(i,j+2) - 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ---------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE TOP RIGHT CORNER (r=R_o,z=H):
i = NX2
j = NY2
r = DBLE(i-1)*dr
SR1_ij  = (-4.0D0*V(i-1,j) + V(i-2,j) + 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ---------------------------------------------------------
! CALCULATING THE SHEAR RATE ON THE TOP LEFT CORNER (r=R_i,z=H):
i = NX1_corner
j = NY2
r = DBLE(i-1)*dr
SR1_ij  = ( 4.0D0*V(i+1,j) - V(i+2,j) - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
! ---------------------------------------------------------
! CALCULATING THE SHEAR RATE AT SMOOTH CORNER (r=[NX1,NX1_corner];
!                                         and z=[NY1_corner,NY1]):
DO k = 1,NY1-NY1_corner+1 ! k=1,11
```

```
  i = NX1 + (k-1)
  j = NY1_corner + (k-1)
  r = DBLE(i-1)*dr
  SR1_ij  = ( 4.0D0*V(i+1,j) - V(i+2,j) - 3.0D0*V(i,j))/(2.0D0*dr) - V(i,j)/r
  SR2_ij  = (-4.0D0*V(i,j-1) + V(i,j-2) + 3.0D0*V(i,j))/(2.0D0*dz)
  SR(i,j) = DSQRT(SR1_ij**2.0D0 + SR2_ij**2.0D0)
END DO
! ------------------------------------------------------------------------------ !
RETURN
END SUBROUTINE SR_PROFILE
! ============================================================================== !
END MODULE WRITE_INFORMATION
! ------------------------------------------------------------------------------ !
```

A.4 GNU GENERAL PUBLIC LICENSE

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those

who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix B

Miscellaneous Theories

B.1 Introduction

In this appendix, some theories are presented that can be related to the issue of Chapter 2. Some of these theories are rather hard to explain and although some significant amount of time and effort has been used in writing the correct idea, the success is only such that the resulting text would simply get in the way of the main objectives of Chapter 2. Also, some of the theories presented here are too "classical" to be presented in the chapter and too important to be left out altogether.

B.2 The Solid- and Continuum Particle

B.2.1 The Solid Particle

When dealing with the motion of a (enormously) large collection of solid particles, the concept of a particle must be redefined. The reason is that it is rather difficult, if not impossible, to gain a solution for multi particle system by looking at every single **solid particle**[1] as the *working* particle of the system. That is, by using[2] Newton's 2^{nd} law $\underline{d\mathbf{p}_{\text{I}}/dt = \mathbf{F}_{\text{I}} + m_{\text{I}}\,\mathbf{g}}$ on every such particle, the following problems arise:

1.A Overpopulation of the solid particles: The number of solid particles, composing the flowing continuum is extremely large and hence the number of governing equations (one equation for each and every solid particle), which need to be solved **simultaneously**, becomes too large for a computer to handle. They need to be solved simultaneously because the velocity $\mathbf{v}_{\text{I}}$ of one solid particle enters as a variable in the function of external surface forces $\mathbf{F}_{\text{I+1}}$ for the neighboring solid particle and visa versa $\mathbf{F}_{\text{I+1}} = \mathbf{F}_{\text{I+1}}(\mathbf{v}_{\text{I}}, \dots) \wedge \mathbf{F}_{\text{I}} = \mathbf{F}_{\text{I}}(\mathbf{v}_{\text{I+1}}, \dots)$.

1.B Whereabouts of the solid particles: For a complex suspension like of concrete, mortar or cement paste, the **exact** material distribution inside it is not known at the beginning of calculation $t = 0$. In other words, it is not known if there is an aggregate, cement grain or water molecule (and so forth) that is occupying the specific coordinates $\mathbf{x}$ at $t = 0$. Since the initial coordinates of

[1] For example, looking at every single aggregate, cement grain, water molecule and so forth.
[2] See below Equation 2.2 (Page 13) for descriptions of variables.

the individual solid particles are not known, it becomes impossible to calculate their coordinates at later times $t > 0$ inside the suspension.

1.C Discontinuity in the velocity function: The individual solid particle will generally travel in a **discontinuous path**[3] due to the collision forces imposed by its neighboring solid particles. This will result in that the velocity $\mathbf{v}_\mathrm{I}$ will only be partially smooth. This lack of smoothness will increase the complexities of the mathematical solution algorithm to be used.

1.D Complexities of the external surface forces: A large problem lies in finding and using the correct function, describing the sum of external surface forces $\mathbf{F}_\mathrm{I} = \mathbf{F}_\mathrm{I}(\mathbf{v}_{\mathrm{I}+1}, \dots)$ applied to a specific solid particle, from its surrounding solid particles. For example, when calculating the velocity $\mathbf{v}_\mathrm{I}$ of a solid particle inside a fresh concrete, the information about the exact mass, shape, surface texture, orientation and so forth, must be known[4] for this solid particle and its surrounding solid particles at the moment of impact. Creating a function of external surface forces $\mathbf{F}_\mathrm{I}$ that uses the above information in a correct manner is difficult enough, but then using it increases the complexities and non-linearity of the governing equation.

B.2.2 The Continuum Particle (CP)

As described in Section 2.2, the continuum particle (CP) consists of large collection of solid particles. An oversimplified illustration of a CP is shown in Figure B.1. This CP is composed of only three solid particles, numbered with the Roman numbers I, II and III. Of course, the CP must consist of a larger number of solid particles, for example to have a smooth velocity, defined by Equation 2.6 (Page 13).

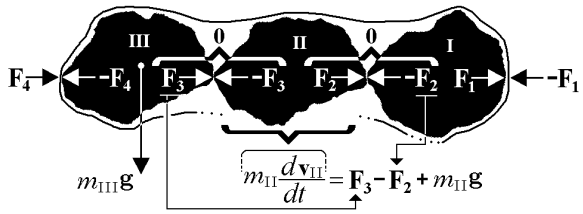


Figure B.1: An oversimplified illustration of a CP. This CP is composed of only three solid particles, numbered with the Roman numbers I, II and III. Each of them comply to Newton's 2nd law: $m_\mathrm{I}\,(d\mathbf{v}_\mathrm{I}/dt) = \mathbf{F}_\mathrm{I} + m_\mathrm{I}\,\mathbf{g} = \mathbf{F}_2 - \mathbf{F}_1 + m_\mathrm{I}\,\mathbf{g}$; $m_\mathrm{II}\,(d\mathbf{v}_\mathrm{II}/dt) = \mathbf{F}_\mathrm{II} + m_\mathrm{II}\,\mathbf{g} = \mathbf{F}_3 - \mathbf{F}_2 + m_\mathrm{II}\,\mathbf{g}$ and $m_\mathrm{III}\,(d\mathbf{v}_\mathrm{III}/dt) = \mathbf{F}_\mathrm{III} + m_\mathrm{III}\,\mathbf{g} = \mathbf{F}_4 - \mathbf{F}_3 + m_\mathrm{III}\,\mathbf{g}$. The summation of these equations, produces the equation of motion for this CP: $d(m_\mathrm{I}\,\mathbf{v}_\mathrm{I} + m_\mathrm{II}\,\mathbf{v}_\mathrm{II} + m_\mathrm{III}\,\mathbf{v}_\mathrm{III})/dt = \mathbf{F}_\mathrm{I} + \mathbf{F}_\mathrm{II} + \mathbf{F}_\mathrm{III} + (m_\mathrm{I} + m_\mathrm{II} + m_\mathrm{III})\,\mathbf{g} = \mathbf{F}_4 - \mathbf{F}_1 + \delta m\,\mathbf{g}$. Only the surface force terms which are applied to the CP from its surroundings, namely $\mathbf{F}_4$ and $-\mathbf{F}_1$, are remaining after this step. The black solid line confining these particles defines the boundary of this CP.

In a complex particle suspension, where the suspended solid particles consist of a broad range in mass, dimension, shape and surface texture, the CP must be large enough to obtain a material and geometric homogeneity. For the case of concrete,

[3] This motion consist of two velocity components $\Rightarrow \mathbf{v}_\mathrm{I} = \mathbf{v} + [\mathbf{v}_\mathrm{I} - \mathbf{v}]$. The first one $\mathbf{v}$ contributes to the overall smooth motion of the particle suspension (see Equation 2.6, Page 13), while the latter $[\mathbf{v}_\mathrm{I} - \mathbf{v}]$ does not so and is of a random nature.

[4] These parameters have effect on the magnitude and direction of external surface forces involved.

mortar or cement paste, then with **material homogeneity** it is meant that the relative material composition inside any CP should be the same as for the suspension as a whole. With **geometric homogeneity** it is meant that the grading[5] inside any CP should be the same as for the overall suspension. With those two homogeneity conditions fulfilled one has ensured identical physical characteristics for all the CPs inside the continuum. With this criteria fulfilled, the dimension of the CP could be considered as $2D_{\max}$, where $D_{\max}$ represents the dimension of the largest solid particle. Now, with the CP-approach, as described with Equation 2.16 (Page 15), the following is gained:

2.A Overpopulation is no longer an issue: When considering the number of equations needed to be solved simultaneously, then contrary to the solid particle-approach, only one equation is necessary in order to gain solution for all the CPs inside the continuum. This is apparent from Equation 2.11 (Page 14).

2.B The issue of the whereabouts is solved: For homogeneous concrete, mortar or cement paste, it is now known what kind of particle is occupying the coordinates $\mathbf{x}$ at the initial time $t = 0$. This is so because every *working* particle (that is, every CP) inside the continuum is now physically identical (at least when the pre-mentioned material and geometric homogeneity conditions are fulfilled). In other words, at $t = 0$, one can randomly pick a spatial point in the continuum and define it as the initial center of mass (CM) coordinates $\mathbf{X}$ of a CP as shown with Equations 2.8 and 2.9 (Page 14).

2.C Smoothness in the velocity function: The individual CP will generally not travel in a discontinuous path, as is the case for the solid particle. As is shown with Equation 2.6 (Page 13), the velocity of the CP is the mass averaged velocity of the entire set of solid particles composing this CP. With this equation, any random and spontaneous contributions from individual solid particles are summarized out in the averaging. Therefore, only the relevant smooth motion of the continuum will remain in the velocity function $\mathbf{v}$ of the CP.

2.D Simplicity of the external surface forces: Since always the same types of solid particles are involved inside and surrounding each CP, the same type of function $\mathbf{F} = \mathbf{F}(\dots)$ can be used on each and every CP. Therefore, only one equation has to be constructed that describes the sum of external surface forces $\mathbf{f} = \mathbf{F}/\delta V$ applied to the CP from its surroundings. The application of $\mathbf{f}$ to the CP, results in a stress state $\boldsymbol{\sigma}$ develops inside the CP. As such, it is possible to relate mathematically $\mathbf{f}$ and $\boldsymbol{\sigma}$, namely with $\mathbf{f}(\mathbf{x}, t) = \nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t)$. This mathematical relationship is the subject of Appendix B.3.

B.3 Calculating External Forces Applied to the CP

In Section 2.2, the equation of motion for the CP was derived. After several steps, the birth of Equation 2.16 (Page 15) was a reality. However, the delivery was not complete since the function $\mathbf{f}$ was not determined. As stated previously, this variable describes the sum of external surface forces applied to the CP from its surroundings. If the terms $\mathbf{F}_{\text{east}}$, $\mathbf{F}_{\text{west}}$, $\mathbf{F}_{\text{top}}$, $\mathbf{F}_{\text{bottom}}$, $\mathbf{F}_{\text{north}}$ and $\mathbf{F}_{\text{south}}$ designate the external surface forces applied to the CP from the corresponding directions, as shown with Figure B.2,

[5] Grading ⇒ Cumulative solid particle size distribution (see Figure 4.4, Page 76).

then $\mathbf{f} = (\mathbf{F}_{\text{east}} + \mathbf{F}_{\text{west}} + \mathbf{F}_{\text{top}} + \mathbf{F}_{\text{bottom}} + \mathbf{F}_{\text{north}} + \mathbf{F}_{\text{south}})/\delta V$. When the CP is subjected to a rate of deformation $\dot{\boldsymbol{\varepsilon}}$ because of the external surface forces, applied to it from its surroundings $\mathbf{f}$, the CP will resist such deformation. This resistance is directly related to a second order tensor equation $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\dot{\boldsymbol{\varepsilon}}, \dots)$ known as the constitutive equation. For many fluids, this equation is represented as $\boldsymbol{\sigma} = -p\,\mathbf{I} + \mathbf{T}$ [9], where the term p is pressure, $\mathbf{I}$ the unit dyadic and $\mathbf{T}$ the extra stress tensor.

To calculate the force $\mathbf{F}_{\text{east}}$ that is applied to the east wall of the CP, then according to the Cauchy's stress principle [72], one has to operate the unit normal vector of this wall, $\mathbf{n} = \mathbf{i}_1$ on the stress state which applies there: $\boldsymbol{\sigma}|_{x_1+\delta x_1/2}$ as is shown with Equation B.1 (see the left illustration of Figure B.2).

$$\frac{\mathbf{F}_{\text{east}}}{\delta x_2 \delta x_3} = \mathbf{i}_1 \cdot \boldsymbol{\sigma}\big|_{x_1+\frac{\delta x_1}{2}} \tag{B.1}$$

Similar to what applies for Equation B.1, to calculate the external surface force $\mathbf{F}_{\text{west}}$ that is applied to the west wall of the CP, one has to operate the unit normal vector of this wall, $-\mathbf{i}_1$ on the stress state which applies there: $\boldsymbol{\sigma}|_{x_1-\delta x_1/2}$. Summarizing that result with Equation B.1 produces the following:

$$\mathbf{F}_{\text{west}} + \mathbf{F}_{\text{east}} = \left[-\mathbf{i}_1 \cdot \boldsymbol{\sigma}\big|_{x_1-\frac{\delta x_1}{2}} + \mathbf{i}_1 \cdot \boldsymbol{\sigma}\big|_{x_1+\frac{\delta x_1}{2}}\right] \delta x_2 \delta x_3 \tag{B.2}$$

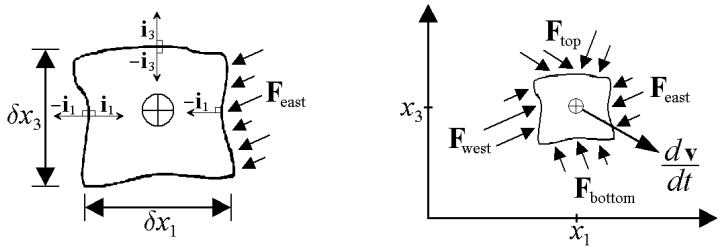


Figure B.2: Forces applied to the CP from its surroundings. The forces applied from the north and south wall are not shown. Comparing the CP in this figure with the oversimplified CP in Figure B.1, then $\mathbf{F}_{\text{west}}$ corresponds to $\mathbf{F}_4$ and $\mathbf{F}_{\text{east}}$ to $-\mathbf{F}_1$.

The stress state of the west and east walls can be calculated (or estimated) with the help of 1st degree Taylor polynomial (see Equation 7.9, Page 157):

$$\boldsymbol{\sigma}\big|_{x_1-\frac{\delta x_1}{2}} = \boldsymbol{\sigma}\big|_{x_1} + \left.\frac{\partial \boldsymbol{\sigma}}{\partial x_1}\right|_{x_1} \left(-\frac{\delta x_1}{2}\right) \tag{B.3}$$

$$\boldsymbol{\sigma}\big|_{x_1+\frac{\delta x_1}{2}} = \boldsymbol{\sigma}\big|_{x_1} + \left.\frac{\partial \boldsymbol{\sigma}}{\partial x_1}\right|_{x_1} \left(\frac{\delta x_1}{2}\right) \tag{B.4}$$

The partial derivative in the above two equations can be looked at as a comparison of stress states between two CPs, placed on either side of the CP in question. They are separated by the distance $2\delta x_1$, relative to their CM:

$$\left.\frac{\partial \boldsymbol{\sigma}}{\partial x_1}\right|_{x_1} = \lim_{\Delta x_1 \to \delta x_1} \frac{\boldsymbol{\sigma}(x_1 + \Delta x_1, x_2, x_3, t) - \boldsymbol{\sigma}(x_1 - \Delta x_1, x_2, x_3, t)}{2\,\Delta x_1} \tag{B.5}$$

Now, putting the two above Taylor approximations (i.e. Equations B.3 and B.4) in Equation B.2, gives the following result:

$$\mathbf{F}_{\text{west}} + \mathbf{F}_{\text{east}} = \mathbf{i}_1 \cdot \left.\frac{\partial \boldsymbol{\sigma}}{\partial x_1}\right|_{x_1} \delta V = \left(\mathbf{i}_1 \frac{\partial}{\partial x_1} \cdot \boldsymbol{\sigma}\right) \delta V \tag{B.6}$$

where the value $\delta x_1 \delta x_2 \delta x_3$ is approximately equal to the volume of the CP, namely equal to δV (see Figure B.2). The forces applied to the south-north wall and to the bottom-top wall of the CP are calculated in the same way as was done for the west-east wall:

$$\mathbf{F}_{\text{south}} + \mathbf{F}_{\text{north}} = \left[-\mathbf{i}_2 \cdot \boldsymbol{\sigma}\big|_{x_2 - \frac{\delta x_2}{2}} + \mathbf{i}_2 \cdot \boldsymbol{\sigma}\big|_{x_2 + \frac{\delta x_2}{2}} \right] \delta x_1 \delta x_3 \tag{B.7}$$

$$\mathbf{F}_{\text{bottom}} + \mathbf{F}_{\text{top}} = \left[-\mathbf{i}_3 \cdot \boldsymbol{\sigma}\big|_{x_3 - \frac{\delta x_3}{2}} + \mathbf{i}_3 \cdot \boldsymbol{\sigma}\big|_{x_3 + \frac{\delta x_3}{2}} \right] \delta x_1 \delta x_2 \tag{B.8}$$

Using the Taylor approximation similar to Equations B.3 and B.4 in the above, produces the following results:

$$\frac{\mathbf{F}_{\text{south}} + \mathbf{F}_{\text{north}}}{\delta V} = \left(\mathbf{i}_2 \frac{\partial}{\partial x_2} \cdot \boldsymbol{\sigma} \right) \quad \wedge \quad \frac{\mathbf{F}_{\text{bottom}} + \mathbf{F}_{\text{top}}}{\delta V} = \left(\mathbf{i}_3 \frac{\partial}{\partial x_3} \cdot \boldsymbol{\sigma} \right) \tag{B.9}$$

Combining Equations B.6 and B.9 gives the sum of external surface forces applied to the CP from its surroundings:

$$\mathbf{f} = \frac{\mathbf{F}}{\delta V} = \left(\mathbf{i}_1 \frac{\partial}{\partial x_1} + \mathbf{i}_2 \frac{\partial}{\partial x_2} + \mathbf{i}_3 \frac{\partial}{\partial x_3} \right) \cdot \boldsymbol{\sigma} = \nabla \cdot \boldsymbol{\sigma} \tag{B.10}$$

B.4 Resolution of the Material Space

Often, the diameter $D_{\max}$ of the largest aggregates used in the concrete mix, is around 16 mm and it can even be as large as 32 mm or higher. In order to secure that all the CPs of the fresh concrete have the same material and geometric homogeneity, as discussed on Page 379, their dimensions can easily be several to tens of centimeters. This situation is considerably different from what, for example applies for water continuum where the size of a CP can be regarded as *infinitely small*[6], or equally as a point item. Because of this fundamental difference, the discussion in the present section is most necessary.

To avoid confusion, it is important to note that everywhere else in this thesis, **except** for here (i.e. Appendix B.4), there is no difference between $\mathbf{x}$ and $\mathbf{r}^{\text{CM}} \equiv \mathbf{x}(\mathbf{X}, t)$, i.e. both of them represents the CM coordinates of a CP. In mathematical terms, this means $\mathbf{x} \equiv \mathbf{r}^{\text{CM}} \equiv \mathbf{x}(\mathbf{X}, t) \ \forall\, t$. In the present appendix, $\mathbf{x}$ might or might not be equal to $\mathbf{x}(\mathbf{X}, t) \equiv \mathbf{r}^{\text{CM}}$.

B.4.1 The CM-Position $\mathbf{r}^{\text{CM}} \equiv \mathbf{x}(\mathbf{X}, t)$ and the Coordinates $\mathbf{x}$

At the first consideration, the price tag of the CP-approach is that the resolution (or sharpness) of the material space has been sacrificed [139]. This is so because with the location $\mathbf{r}^{\text{CM}} \equiv \mathbf{x}(\mathbf{X}, t)$, it is referred to the CM-position of a CP, consisting of the finite sized volume ($\delta V \approx \delta x_1\, \delta x_2\, \delta x_3$) as shown with Figure 2.2 (Page 12), and not of some point item at the same location. The dimensions δx_1, δx_2 and δx_3 of the CP, represent therefore the highest resolution one can *squeeze* out of the physical continuum. Hence the term Δx_3 in Equation 2.21 (Page 17) and Δx_1 in Equation B.5

[6] With *infinitely small* it is meant that the diameter $D_{\max}$ of the largest solid particle that is suspended in the continuum, divided with the characteristic thickness of the flow D_{flow}, is approximately zero: $D_{\max}/D_{\text{flow}} \to 0$. In this perspective, the term *small* is also dependent on how large the flowing system is. For example, our **Galaxy** can be regarded as infinitely small when modeling the whole **Universe** as a self gravitating fluid (see for example textbook by Battaner [11]).

did not approach zero but rather approached the smallest dimension available to the continuum, namly δx_1 and δx_3.

When one is referring to some coordinates in relation to rheology, for example when utilizing Equation 2.17 (Page 16), it is explicitly meant the CM-position $\mathbf{x}(\mathbf{X}, t)$ of a CP, which consist of a finite sized volume δV. When talking separately about some coordinates $\mathbf{x}$ of the coordinate axes, without giving it a correlation to any governing equation of continuum mechanics, their values can be whatever one chooses it to be. It has no physical restrictions and is in that sense a completely independent variable. When solving Equation 2.17, one is forced to treat the dependent variable $\mathbf{x}(\mathbf{X}, t)$ as an independent one, namely as $\mathbf{x}$. In other words, it must be assumed that at every given moment of time t, the specific coordinates $\mathbf{x} = (x_1, x_2, x_3)$ always represents a CM-position of a CP. As shown in the left illustration of Figure B.3, this is not so. Because of the finite sized volume δV, only a few locations $\mathbf{x}$ actually represents a CM-position of a CP, designated with $\mathbf{x}^{\mathrm{I}}$, $\mathbf{x}^{\mathrm{II}}$ and $\mathbf{x}^{\mathrm{III}}$.

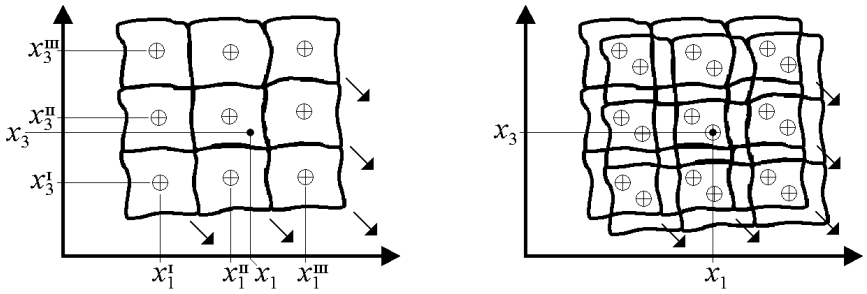


Figure B.3: The arrows represents the direction of flow. To the left: Visual demonstration of the characteristic difference between $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{x}(\mathbf{X}, t) \in [\mathbf{x}^{\mathrm{I}}, \mathbf{x}^{\mathrm{II}}, \mathbf{x}^{\mathrm{III}}]$. The latter coordinates always represent the CM coordinates of a CP, whereas the former only do so occasionally. To the right: By utilization of the MSP it becomes more likely that the arbitrary point $\mathbf{x}$ represents a CM coordinates $\mathbf{x}(\mathbf{X}, t)$ of a CP.

In the cases when the coordinates $\mathbf{x}$ in Equation 2.17 do not represent the CM-position $\mathbf{x}(\mathbf{X}, t)$ of a CP, this equation represents no longer the Newton's 2nd law at that same location, as outlined with the CP-approach in Section 2.2. In fluid mechanics of water, this type of considerations is redundant since the size of the CP is, from practical point of view, *infinitely* small and therefore can be regarded as a point item. Therefore, in that case the arbitrary coordinates $\mathbf{x}$ will (more or less) always represent a CM-position $\mathbf{x}(\mathbf{X}, t)$ of a CP. When this latter condition applies, the governing Equation 2.17 can be related to the coordinates $\mathbf{x}$, rather than to a physical entity like of the CP.

B.4.2 Material Superposition Principle (MSP)

When considering Equations 2.11 and 2.12, the first impression is that the CPs arrange themselves in an orderly fashion as shown with the left illustration of Figure B.3. This was assumed when estimating the derivatives in Equations 2.21 and B.5. However, there are no restrictions by introducing additional CPs into the continuum as shown with the right illustration of Figure B.3. Because the mass of the continuum may not increase in the process, the original CPs must share their material with the new additional CPs. In other words, every CP inside the continuum must share its

material with its neighboring CPs. From a mathematical point of view, nothing has changed in the process. When defining a CP by the summation shown in Equation 2.3 (Page 13), it consists of the same number N of solid particles. However, some of these solid particles will also be used when defining its neighboring CP. This setup will be known here as the MATERIAL SUPERPOSITION PRINCIPLE (MSP). The benefits of this principle is that one can increase the resolution of the material space. That is, increase the possibility that, at any given moment of time, the arbitrary coordinates $\mathbf{x}$ actually represents the CM-position of a CP, namely $\mathbf{r}^{\mathrm{CM}} \equiv \mathbf{x}(\mathbf{X}, t)$. Applying the MSP to Equation 2.21, results in that Δx_3 does not have to approach δx_3, but can rather start to approach zero: $\Delta x_3 \to 0$. This is demonstrated with Figure B.4.

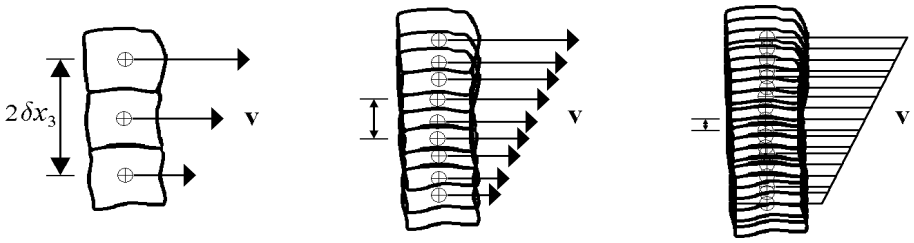
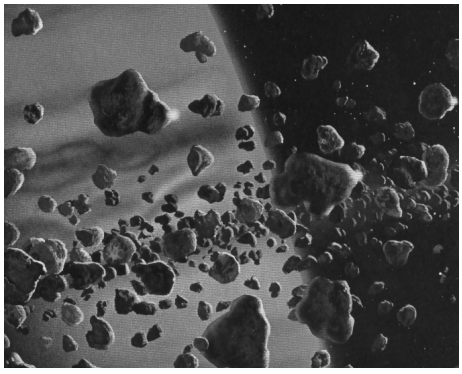


Figure B.4: Graphical presentation of the circumstances for Equation 2.21 (Page 17). To the left: Without the MSP, then $\Delta x_3 \to \delta x_3$. In the middle and to the right: By utilization of the MSP, then $\Delta x_3 \to 0$. Similar considerations can be made, in relation to Equation B.5.

The purpose of the CP is to provide a smooth velocity $\mathbf{v}$, calculated with Equation 2.18 and defined by Equation 2.6. However, the purpose of the MSP is to provide smooth velocity $\mathbf{v}$ at all spatial points $\mathbf{x}$ in the continuum Ω.

B.5 Rings of Saturn as Fluid

Utilization of Equation 2.17 on the rings of Saturn (Figure B.5) will now commence. The rings are mostly made of ice fragments (solid particles), ranging from few centimeters to few meters across, but there are also some traces of silicate and carbon minerals, indicating that rock fragments are also present [3]. Using Equation 2.17 on these set of solid particles, is done to stress out that the fluid approach is in essence the theory about the collective motion of a very large number of solid particles (see figure to the right[7]), without going into the detailed motion of every single one of them. The largest solid particles are treated in the exactly the same manner as the smallest ones.



As the rings are only around Saturn's equator, the cylindrical coordinate system will be used. The z-direction (or equally, the x_3-direction) will be pointing in the axis of symmetry as shown in Figure B.5 to the right. Starting with the most general

[7] An artist rendition of the rings at close range [54].

velocity field $\mathbf{v} = v_{\mathrm{r}}(r,\theta,z,t)\,\mathbf{i}_{\mathrm{r}} + v_{\theta}(r,\theta,z,t)\,\mathbf{i}_{\theta} + v_{\mathrm{z}}(r,\theta,z,t)\,\mathbf{i}_{\mathrm{z}}$, one can at once discard any z-dependencies since the fluid flow is only around the equator. Also, since all the CPs are flowing in a circular orbit, it is possible to drop out the v_{r}-function. After these steps, the general velocity field has reduced to $\mathbf{v} = v_{\theta}(r,\theta,t)\,\mathbf{i}_{\theta}$. Furthermore, since the individual CP (or equally, the individual rock/ice fragments composing the CP) is neither accelerating nor de-accelerating in the θ-direction, a θ-independence can be assumed. In accordance with the above text, the velocity profile of the rings will be given by Equation B.11.

$$\mathbf{v} = v_{\theta}(r,t)\,\mathbf{i}_{\theta} \tag{B.11}$$

Figure B.5: It is assumed that the velocity profile of the rings of Saturn is $\mathbf{v} = v_{\theta}(r,t)\,\mathbf{i}_{\theta}$. To the left: An artist rendition of the Cassini spacecraft over the rings [60]. To the right: A schematic top view of the rings and the planet.

Since the individual rock/ice fragments are generally not colliding with each other, the CPs are neither interacting with each other[8] (meaning $\mathbf{f} = \nabla\cdot\boldsymbol{\sigma} = 0$) and hence the term $\nabla\cdot\boldsymbol{\sigma} + \rho\,\mathbf{g}$ will consists solely of the gravitational force $\rho\,\mathbf{g} = -\hat{G}M\rho/r^2\;\mathbf{i}_{\mathrm{r}}$. The terms $\hat{G}$ and M are the gravitational constant and the mass of Saturn, respectively. Using $\nabla\cdot\boldsymbol{\sigma} + \rho\,\mathbf{g} = -\hat{G}M\rho/r^2\;\mathbf{i}_{\mathrm{r}}$ and the above velocity profile $\mathbf{v} = v_{\theta}(r,t)\,\mathbf{i}_{\theta}$ in Equation 2.17 gives, after some derivations, Equations B.12 and B.13.

$$\rho\frac{\partial v_{\theta}(r,t)}{\partial t} = 0 \qquad \Rightarrow \quad \mathbf{v} = v_{\theta}(r)\,\mathbf{i}_{\theta} \tag{B.12}$$

$$\rho\frac{v_{\theta}^2(r)}{r} = \frac{\hat{G}M\rho}{r^2} \quad \Rightarrow \quad v_{\theta}(r) = \sqrt{\frac{\hat{G}M}{r}} \tag{B.13}$$

From Equation B.12 the velocity gradient tensor $\nabla\mathbf{v}$ can be calculated, which through Equation 2.20 (Page 17), gives the strain rate tensor which applies in the rings:

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\frac{dv_{\theta}(r)}{dr} - \frac{v_{\theta}(r)}{r}\right)(\mathbf{i}_{\mathrm{r}}\mathbf{i}_{\theta} + \mathbf{i}_{\theta}\mathbf{i}_{\mathrm{r}}) \tag{B.14}$$

Using the above result in Equation 2.24 (Page 18) with the concomitant use of Equation B.13, gives the shear rate which applies inside the rings, shown with Equation B.15. By using values from a textbook by Arny [3], the mass of Saturn and its

[8] From Equation 2.25 (Page 26) when no collisions are occuring between the soldi particles ($\tilde{N}_{\mathrm{dc}} = \tilde{N}_{\mathrm{ic}} = 0$), the shear viscosity becomes zero ($\eta = 0$). Fluids with zero viscosity are termed as **superfluid** [108], and their main characteristics is that they flow continuously without stopping by their own friction (because there are none). A known superfluid, besides the rings of Saturn, is Helium in liquid state cooled below 2.17K at atmospheric pressure [108].

radius is given by $M = 5.69\,10^{26}$kg and $r = 60268\,$km respectively. By using the photograph in Figure B.6 [60], the visual part of the rings start roughly at $R_{\mathrm{i}} = 76400\,$km from the center of Saturn, and ends approximately at $R_{\mathrm{o}} = 146000\,$km. The gravitational constant is $\hat{G} = 6.6726 \cdot 10^{-11}\mathrm{Nm}^2/\mathrm{kg}^2$. Using these values in Equations B.13 and B.15 produces the velocity profile $\mathbf{v} = v_\theta(r)\,\mathbf{i}_\theta$ and the shear rate profile $\dot{\gamma} = \dot{\gamma}(r)$ shown in Figure B.6. The velocity values shown in the figure, is in accordance with Doppler-shift data [153].

$$\dot{\gamma} = \left| \frac{dv_\theta(r)}{dr} - \frac{v_\theta(r)}{r} \right| = \frac{3}{2} \frac{\sqrt{\hat{G}M}}{r^{3/2}} \tag{B.15}$$

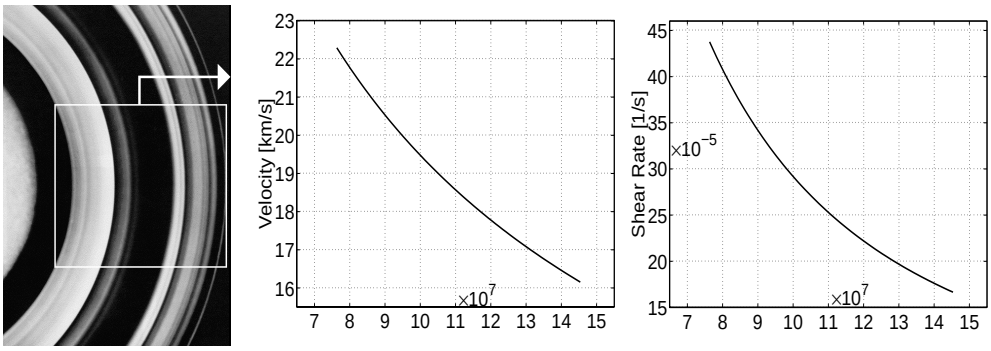


Figure B.6: To the left: Partial top view of Saturn and its rings [60]. Center: The velocity profile $\mathbf{v} = v_\theta(r)\,\mathbf{i}_\theta$ of the rings. To the right: The shear rate profile $\dot{\gamma} = \dot{\gamma}(r)$ of the rings.

B.6 Navier-Stokes Equation

Assuming a constant[9] shear viscosity $\eta = \mu$ in an incompressible fluid ($d\rho/dt = -\rho\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}) = 0$), then from Equations 2.20 and 2.23 (Page 17), the divergence of the extra stress tensor becomes:

$$\begin{aligned} \nabla \cdot \mathbf{T} &= \nabla \cdot (2\,\mu\,\dot{\boldsymbol{\varepsilon}}(\mathbf{x},t)) = \mu\,\left[\nabla \cdot \nabla\mathbf{v} + \nabla \cdot (\nabla\mathbf{v})^{\mathrm{T}}\right] = \\ &= \mu\,\left[\mathbf{i}_{\mathrm{q}} \frac{\partial}{\partial x_{\mathrm{q}}} \cdot \frac{\partial v_{\mathrm{p}}}{\partial x_{\mathrm{k}}} \mathbf{i}_{\mathrm{p}}\mathbf{i}_{\mathrm{k}} + \mathbf{i}_{\mathrm{q}} \frac{\partial}{\partial x_{\mathrm{q}}} \cdot \frac{\partial v_{\mathrm{k}}}{\partial x_{\mathrm{p}}} \mathbf{i}_{\mathrm{p}}\mathbf{i}_{\mathrm{k}}\right] = \\ &= \mu\,\left[\frac{\partial}{\partial x_{\mathrm{p}}} \frac{\partial v_{\mathrm{p}}}{\partial x_{\mathrm{k}}} \mathbf{i}_{\mathrm{k}} + \frac{\partial}{\partial x_{\mathrm{p}}} \frac{\partial v_{\mathrm{k}}}{\partial x_{\mathrm{p}}} \mathbf{i}_{\mathrm{k}}\right] = \mu\,\left[\frac{\partial}{\partial x_{\mathrm{k}}} \frac{\partial v_{\mathrm{p}}}{\partial x_{\mathrm{p}}} \mathbf{i}_{\mathrm{k}} + \frac{\partial^2 v_{\mathrm{k}}}{\partial x_{\mathrm{p}}\,\partial x_{\mathrm{p}}} \mathbf{i}_{\mathrm{k}}\right] \end{aligned} \tag{B.16}$$

In the above, the *indicial notation* [74, 72] in Cartesian coordinate system, is used. Going back to vector notation, the above can be rewritten as:

$$\nabla \cdot \mathbf{T} = \mu\,[\nabla(\nabla \cdot \mathbf{v}) + \nabla \cdot \nabla\mathbf{v}] = \mu\nabla^2\mathbf{v} \tag{B.17}$$

From the above result, the divergence of the stress tensor $\nabla \cdot \boldsymbol{\sigma} = \nabla \cdot (-p\,\mathbf{I} + \mathbf{T})$ will be equal to $-\nabla p + \mu\nabla^2\mathbf{v}$. Putting this result in Equation 2.17 produces the

[9] For a given temperature and pressure, a fluid with constant shear viscosity η is designated as a Newtonian fluid.

Navier-Stokes equation for incompressible fluid, shown with Equation B.18. The viscosity term $\nu = \mu/\rho$ is known as the **kinematic viscosity** [72].

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g} \tag{B.18}$$

B.7 Energy and Work

The main objectives of this appendix is to generate the energy equation for the CP. In doing so the material volume $V(t)$ [43] is used as a starting point, rather than the CP. To begin with, a description of the material volume is made with the concomitant re-derivation of Equation 2.17 to demonstrate the validity of this approach. As will be shortly apparent, the material volume approach is much simpler than what applied previously with the direct CP-approach. However, the first mentioned approach is more descriptive of what fluid actually consist of.

B.7.1 Material Volume

Consider a large collection of CPs enclosed in a deformable volume. During calculation time, this volume always consist of the same CPs, and hence has always the same amount of mass $m = \sum \delta m = \text{constant}$, c.f. Equation 2.1 (Page 12). This volume is usually designated as the **material volume** [43].

Conservation of Mass

The mass of the material volume can be gained by integrating the density ρ (of a CP) (see Equation 2.14, Page 15) over its volume $V(t)$ as shown with Equation B.19.

$$m = \iiint\limits_{V(t)} \rho \, dV \tag{B.19}$$

The term dV represents the resolution of the material space and with the MSP, this resolution is much larger than presented with the volume δV of the CP; i.e. the number of non-overlapping CPs inside the material volume $V(t)$ is $n_{\text{CP}} = V(t)/\delta V$ (see the left illustration of Figure B.3), while the total number of CPs inside it is $n_{\text{CP}}^{\text{MSP}} = V(t)/dV$ (see the right illustration of Figure B.3). As such, the summation in Equation B.19 is from 1 to $n_{\text{CP}}^{\text{MSP}}$.

Taking the time derivative[10] of Equation B.19, with the $m = \text{constant}$ condition in mind, gives the following:

$$\begin{aligned} \frac{dm}{dt} &= \frac{d}{dt}\iiint\limits_{V(t)} \rho \, dV = \iiint\limits_{V(t)} \frac{\partial \rho}{\partial t} \, dV + \iint\limits_{\partial V(t)} \rho \, \mathbf{v} \cdot \mathbf{n} \, dS \\ &= \iiint\limits_{V(t)} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \, \mathbf{v}) \right) dV = 0 \end{aligned} \tag{B.20}$$

In the above equation, the **Leibnitz theorem** [20, 93] was first applied and then the **integral theory of Gauss** [20, 93]. Note that the velocity term $\mathbf{v}$ that appears after applying the theorem of Leibnitz, is the velocity of the boundary of the material

[10] See Footnote 10 (Page 15) about the time derivative d/dt.

volume $\partial V(t)$, and therefore is in this case the velocity of the CPs located there. Finally, since the domain of integration $V(t)$ is not fixed, **duBois-Reymonds** lemma can be applied to conclude that the integrand in the above equation must vanish [71]. After this step, the well-known equation of continuity is produced:

$$\frac{d\rho}{dt} + \rho \operatorname{tr}(\dot{\boldsymbol{\varepsilon}}) = 0 \quad \text{or equally} \quad \frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad \text{since} \quad \operatorname{tr}(\dot{\boldsymbol{\varepsilon}}) \equiv \nabla \cdot \mathbf{v} \tag{B.21}$$

The name *continuity equation* for the above, is given because this equation assumes that the velocity and density are defined in every point in space [93]. This assumption is in most cases valid for a coarse particle suspension, like the fresh concrete, and then through the MSP (see the last paragraph in Appendix B.4.2).

Conservation of Momentum

Newton's 2nd law can be applied to the material volume $V(t)$ as shown with Equation B.22. The term $\mathbf{p}_{V(t)}$ is the momentum of the material volume and $\mathbf{F}_{V(t)}$ is the sum of external surface forces applied on it from its surroundings as shown with Equation B.23.

$$\frac{d\mathbf{p}_{V(t)}}{dt} = \mathbf{F}_{V(t)} \tag{B.22}$$

$$\mathbf{p}_{V(t)} = \iiint\limits_{V(t)} \rho\, \mathbf{v}\, dV \quad \wedge \quad \mathbf{F}_{V(t)} = \iint\limits_{\partial V(t)} \mathbf{t}\, dS + \iiint\limits_{V(t)} \rho\, \mathbf{g}\, dV \tag{B.23}$$

The term $\mathbf{t} = \mathbf{n} \cdot \boldsymbol{\sigma}$ in the above, is sometimes named **traction** [72] and describes the force per unit area ($[\mathrm{N/m^2}]$) applied at the boundary $\partial V(t)$, from the outer surroundings of the material volume. The unit normal vector $\mathbf{n}$ is located at the boundary $\partial V(t)$ and points out away from the material volume, c.f. the Cauchy's stress principle [72]. Combining Equations B.22 and B.23 and then using the Leibnitz theorem, produces Equation B.24.

$$\iiint\limits_{V(t)} \frac{\partial(\rho\, \mathbf{v})}{\partial t}\, dV + \iint\limits_{\partial V(t)} (\rho\, \mathbf{v}) \mathbf{v} \cdot \mathbf{n}\, dS = \iint\limits_{\partial V(t)} \mathbf{n} \cdot \boldsymbol{\sigma}\, dS + \iiint\limits_{V(t)} \rho\, \mathbf{g}\, dV \tag{B.24}$$

Using the integral theory of Gauss, lemma of duBois-Reymonds and the result from Equation B.21 in the above, gives after some rearrangement[11] the following:

$$\rho \frac{d\mathbf{v}}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho\, \mathbf{g} \tag{B.25}$$

The above equation is identical to Equation 2.17 (Page 16).

B.7.2 Conservation of Energy

Mechanical Energy Equation

Gravity $\mathbf{g}$ $[\mathrm{m/s^2}]$ can be represented in terms of the gradient of a potential function Φ $[\mathrm{m^2/s^2}]$, called the **geopotential**, as shown with the equation below [48].

$$\mathbf{g} = -g\, \mathbf{i}_z = -\nabla \Phi \tag{B.26}$$

[11] $\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho\, \mathbf{v}\, \mathbf{v}) = \frac{\partial(\rho \mathbf{v})}{\partial t} + \mathbf{v}\, \mathbf{v} \cdot \nabla \rho + \rho\, \mathbf{v} \cdot \nabla \mathbf{v} + \rho\, \mathbf{v}\, \nabla \cdot \mathbf{v} = \rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) + \mathbf{v}(\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho) + \mathbf{v}\, \rho\, \nabla \cdot \mathbf{v} = \rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) + \mathbf{v} \frac{d\rho}{dt} + \mathbf{v}\, \rho\, \nabla \cdot \mathbf{v} = \rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) + \mathbf{v}\, (\frac{d\rho}{dt} + \rho\, \nabla \cdot \mathbf{v}) = \rho\, \frac{dv}{dt}$

Since the geopotential Φ is only dependent[12] on elevation z, i.e. $\Phi = -\hat{G}Mz^{-1}$, the following maneuver can be concluded: $\mathbf{v}\cdot\nabla\Phi = \partial\Phi/\partial t + \mathbf{v}\cdot\nabla\Phi = d\Phi/dt$. Multiplying Equation B.25 with the velocity $\mathbf{v}$ and then using the above maneuver, followed by some other mathematical steps[13], the mechanical energy[14] equation [20] is produced:

$$\rho\frac{d}{dt}\left(\frac{\mathbf{v}\cdot\mathbf{v}}{2}+\Phi\right) = (\nabla\cdot\boldsymbol{\sigma})\cdot\mathbf{v} \tag{B.27}$$

Kinetic and Internal Energy

The sum of the kinetic K [J] and internal U [J] energy of the material volume $V(t)$, is given by the following:

$$K+U = \frac{1}{2}\iiint\limits_{V(t)} \rho\,\mathbf{v}\cdot\mathbf{v}\,dV + \iiint\limits_{V(t)} \rho\,u\,dV \tag{B.28}$$

where $u = u(\mathbf{x},t)$ [J/kg] is the internal energy (per unit mass) of a CP (labeled with $\mathbf{X}$) that is passing through the position $\mathbf{x} = \mathbf{x}(\mathbf{X},t)$ at the time t. Of course, $\mathbf{x}$ must be located somewhere inside the material volume $V(t)$. Taking the time derivative of the above, Equation B.29 is produced after some mathematical steps.

$$\frac{d(K+U)}{dt} = \frac{1}{2}\iiint\limits_{V(t)}\left[\rho\,\frac{d[\frac{\mathbf{v}\cdot\mathbf{v}}{2}+u]}{dt} + \left[\frac{\mathbf{v}\cdot\mathbf{v}}{2}+u\right]\left(\frac{d\rho}{dt}+\rho\,\nabla\cdot\mathbf{v}\right)\right]dV \tag{B.29}$$

The last term in the above integrand must vanish in accordance with Equation B.21.

Rate of Heat Supply

The rate of heat supply $\dot{Q}$ [J/s], to the material inside the material volume $V(t)$, is due to two factors. First, it is the rate of heat $\dot{h} = \dot{h}(\mathbf{x},t)$ [J/(kg · s)] generated or lost per unit mass inside the material volume. For example, heat is constantly generated inside the cement based material $\dot{h} > 0$ due to the hydration of the cement clinker as shown with the right illustration of Figure 2.15 (Page 35).

The second factor is due to heat flux (or heat conduction [20]) $\mathbf{q}$ [J/(m^2 · s)] through the boundary of the material volume $\partial V(t)$. This vector points in the direction[15] of heat flow. Expressing the above text in terms of mathematics, Equation B.30 is generated.

$$\dot{Q} \equiv \frac{\cancel{dQ}}{dt} = \iiint\limits_{V(t)} \rho\,\dot{h}\,dV - \iint\limits_{\partial V(t)} \mathbf{q}\cdot\mathbf{n}\,dS = \iiint\limits_{V(t)}\left(\rho\,\dot{h} - \nabla\cdot\mathbf{q}\right)dV \tag{B.30}$$

[12] $\hat{G}$ and M are the gravitational constant and the mass of Earth, respectively.

[13] I) $\frac{d(\mathbf{v}\cdot\mathbf{v})}{dt} = \frac{d\mathbf{v}}{dt}\cdot\mathbf{v} + \mathbf{v}\cdot\frac{d\mathbf{v}}{dt} = 2\,\mathbf{v}\cdot\frac{d\mathbf{v}}{dt} \Rightarrow \mathbf{v}\cdot\frac{d\mathbf{v}}{dt} = \frac{d}{dt}\left(\frac{\mathbf{v}\cdot\mathbf{v}}{2}\right)$. II) $\mathbf{v}\cdot(\nabla\cdot\boldsymbol{\sigma}) = (\nabla\cdot\boldsymbol{\sigma})\cdot\mathbf{v}$.

[14] With sufficiently small $\nabla\cdot\dot{\boldsymbol{\varepsilon}}$, the term $(\nabla\cdot\mathbf{T})\cdot\mathbf{v}$ becomes much smaller than other relevant terms in Equation B.27. Also assuming a steady state fluid flow ($\partial p/\partial t = 0$), the following applies: $(\nabla\cdot\boldsymbol{\sigma})\cdot\mathbf{v} \approx -\mathbf{v}\cdot\nabla p = dp/dt$. With g = constant near the Earth surface when integrating Equation B.26, one have $\Phi = \Phi_o + g(z - z_o)$, where Φ_o and z_o are the geopotential and elevation at some reference point. Putting all these results in Equation B.27 and assuming incompressibility, produces $\frac{d}{dt}(\mathbf{v}\cdot\mathbf{v}/2 + z\,g + p/\rho) = 0$. This equation is known as the **Bernoulli equation** [72, 106] and states that the quantity $\mathbf{v}\cdot\mathbf{v}/2 + z\,g + p/\rho$ does not change following a CP, or $\mathbf{v}\cdot\mathbf{v}/2 + z\,g + p/\rho =$ constant. This constant is generally different for one CP to the next. If the same constant is to apply for the whole continuum, the flow has to be also irrotational: $\nabla\times\mathbf{v} = 0$ [72].

[15] At hot summer day, the surroundings will supply heat to the concrete mix and hence $\mathbf{q}\cdot\mathbf{n} < 0$, while at cold winter day, the surroundings will extract heat from it: $\mathbf{q}\cdot\mathbf{n} > 0$. As stated previously, the unit normal vector $\mathbf{n}$ is located at the boundary $\partial V(t)$ and points out from the material volume.

In the above equation, the integral theory of Gauss [20, 93] was applied. The heat flux $\mathbf{q}$ is given by the Fourier law: $\mathbf{q} = -\mathbf{k} \cdot \nabla T = -k \nabla T$, where $\mathbf{k}$ is the tensor of thermal conductivity, equal to $\mathbf{k} = k\,\mathbf{I}$ for isotropic material [43]. The variable T is the temperature.

The Newton's law of heat convection $\mathbf{q}_\mathrm{N} = \hat{h}(T - T_\mathrm{o})\mathbf{n}$ [47] is often combined with the Fourier's law of heat conduction, to provide the Robin boundary condition: $\mathbf{q} \cdot \mathbf{n} = \mathbf{q}_\mathrm{N} \cdot \mathbf{n} \Rightarrow \partial T/\partial n \equiv \nabla T \cdot \mathbf{n} = \hat{h}\,(T_\mathrm{o} - T)/k$. The terms $\hat{h}$ and T_o are the convection heat transfer coefficient per unit area and the temperature just outside the boundary $\partial V(t)$, respectively. Here, the variable T is the temperature **at** the boundary of the material volume $\partial V(t)$.

Rate of Work (Power of the Forces f and $\rho\,\mathbf{g}$)

The rate in mechanical effort (or rate of work) conducted on the material volume $V(t)$, from its surroundings, is designated with $\dot{W}$ [J/s] and is given by Equation B.31. The term $\mathbf{g} \cdot \mathbf{v}$ expresses the rate of gravitational work done on the CP. The term[16] $\mathbf{t} \cdot \mathbf{v}$ represents the (viscous) rate of work applied on the CP, from its surroundings.

$$\dot{W} \equiv \frac{\cancel{dW}}{dt} = \iiint\limits_{V(t)} \rho\,\mathbf{g} \cdot \mathbf{v}\,dV + \iint\limits_{\partial V(t)} \mathbf{t} \cdot \mathbf{v}\,dS \tag{B.31}$$

It is usually the rate of gravitational work $\iiint \rho\,\mathbf{g} \cdot \mathbf{v}\,dV \geq 0$ that will make the concrete flow inside the mold or formwork, while it is the rate of viscous work $\iint \mathbf{t} \cdot \mathbf{v}\,dS \leq 0$ that will slow down this process. Now, applying the integral theory of Gauss [20, 93] on Equation B.31, gives the equation shown just below. In this process, the term $\mathbf{t} \cdot \mathbf{v}$ is transformed to $\nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{v})$ and is equal[17] to $(\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{v} + \boldsymbol{\sigma} : \nabla \mathbf{v}$.

$$\dot{W} = \iiint\limits_{V(t)} \left(-\rho\,\mathbf{v} \cdot \nabla \Phi + \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{v})\right) dV \tag{B.32}$$

The integrand in the above equation, is equal to $[-\rho\,d\Phi/dt + (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{v}] + \boldsymbol{\sigma} : \nabla \mathbf{v}$. It is apparent from Equation B.27, that the first part of this result, is equal to $d(\mathbf{v} \cdot \mathbf{v}/2)/dt$. As such, that relationship describes the rate of work, given by the surroundings and used in supplying or extracting kinetic energy to or from the CP. In other words, the velocity $\mathbf{v}$ of the CP, increases and decreases due three factors: I) Due to the surrounding viscous shearing $(\nabla \cdot \mathbf{T}) \cdot \mathbf{v}$. II) Due to the difference in surrounding pressure $(-\nabla p) \cdot \mathbf{v}$ and III) Due to its change in elevation $d\Phi/dt$. As shown with Equation B.39 shortly, the second part of the above result, $\boldsymbol{\sigma} : \nabla \mathbf{v}$, is equal to $\eta\,\dot{\gamma}^2$, and therefore it can be regarded as the rate of work, used by the surroundings, in continuously deforming the CP. From the above, Equation B.31 can be rewritten to:

$$\dot{W} = \iiint\limits_{V(t)} \rho \frac{d}{dt}\left(\frac{\mathbf{v} \cdot \mathbf{v}}{2}\right) dV + \iiint\limits_{V(t)} \eta\,\dot{\gamma}^2\,dV \tag{B.33}$$

[16] $\mathbf{t} \cdot \mathbf{v} = (\mathbf{n} \cdot \boldsymbol{\sigma}) \cdot \mathbf{v} = (\boldsymbol{\sigma} \cdot \mathbf{v}) \cdot \mathbf{n}$.

[17] $\nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{v}) = \mathbf{i}_\mathrm{p} \frac{\partial}{\partial x_\mathrm{p}} \cdot \left(\sigma_\mathrm{ij} \mathbf{i}_\mathrm{i} \mathbf{i}_\mathrm{j} \cdot v_\mathrm{k} \mathbf{i}_\mathrm{k}\right) = \mathbf{i}_\mathrm{p} \frac{\partial}{\partial x_\mathrm{p}} \cdot \left(\sigma_\mathrm{ij} v_\mathrm{j} \mathbf{i}_\mathrm{i}\right) = \mathbf{i}_\mathrm{p} \cdot \frac{\partial \sigma_\mathrm{ij}}{\partial x_\mathrm{p}} v_\mathrm{j} \mathbf{i}_\mathrm{i} + \mathbf{i}_\mathrm{p} \cdot \sigma_\mathrm{ij} \frac{\partial v_\mathrm{j}}{\partial x_\mathrm{p}} \mathbf{i}_\mathrm{i} = \frac{\partial \sigma_\mathrm{ij}}{\partial x_\mathrm{i}} v_\mathrm{j} + \sigma_\mathrm{ij} \frac{\partial v_\mathrm{j}}{\partial x_\mathrm{i}} = (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{v} + \boldsymbol{\sigma} : \nabla \mathbf{v}$. This is so since $\nabla \cdot \boldsymbol{\sigma} = \mathbf{i}_\mathrm{r}(\partial/\partial x_\mathrm{r}) \cdot \sigma_\mathrm{ij} \mathbf{i}_\mathrm{i} \mathbf{i}_\mathrm{j} = (\partial \sigma_\mathrm{ij}/\partial x_\mathrm{i})\,\mathbf{i}_\mathrm{j}$. The operator **:** is called *double dot* [74]. The general formula is $A_\mathrm{ij} B_\mathrm{ij} = \mathbf{A} : \mathbf{B}$ and the outcome of this product is a scalar, just as the outcome between two vectors operated with the single dot operator, is a scalar.

Defining $\dot{w}$ [J/(kg · s)] as the rate of work applied on a CP from its surroundings and then by summation of all these work components with an integration, gives $\dot{W} = \iiint \rho\, \dot{w}\, dV$. Using this concept in the above equation, with the concomitant use of lemma duBois-Reymonds [71], gives the following equation:

$$\dot{w} = \frac{d}{dt}\left(\frac{\mathbf{v}\cdot\mathbf{v}}{2}\right) + \frac{\eta\,\dot{\gamma}^2}{\rho} \tag{B.34}$$

First Law of Thermodynamics

The first law of thermodynamics states that in a system of constant mass, then energy cannot be created or destroyed [20, 49, 108]. This philosophy can be stated with the following *time rated* equation:

$$\frac{d(U+K)}{dt} = \frac{đQ}{dt} + \frac{đW}{dt} \tag{B.35}$$

The sign convention in the above, is such that energy, heat and work which is transfer **into** the material volume $V(t)$ is **positive**: $d(U+K) > 0$, $đQ > 0$ and $đW > 0$. By writing $đQ$ or $đW$ instead of dQ or dW is done do emphases that those variables are not a perfect differentials [49, 72]. Generalizing from many experimental observations, it is shown that when a system is carried through a cycle and returned to its initial state, then $\oint đQ \neq 0$ and $\oint đW \neq 0$ [72]. In other words, it is not possible to speak of the work content or the heat content of the system at any one particular time. Work and heat are not state function or system properties. They exist only in the form of energy being transferred and have no individual identities in the system [72]. However, experience shows that $\oint đQ + \oint đW = 0$ [72].

Now, Combining Equations B.29, B.30, B.31 and B.35 and then applying lemma of duBois-Reymonds [71], the total energy equation for the CP is produced:

$$\rho\frac{d}{dt}\left(\frac{\mathbf{v}\cdot\mathbf{v}}{2} + \Phi + u\right) = \rho\,\dot{h} - \nabla\cdot q + (\nabla\cdot\boldsymbol{\sigma})\cdot\mathbf{v} + \boldsymbol{\sigma} : \nabla\mathbf{v} \tag{B.36}$$

This is the total energy equation for a CP (labeled with $\mathbf{X}$) that is passing through the position $\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$ at the time t.

Internal Energy

By subtracting Equation B.27 from Equation B.36, the equation for **internal energy** of the CP is produced, shown below.

$$\rho\frac{du}{dt} = \rho\,\dot{h} - \nabla\cdot q + \boldsymbol{\sigma}{:}\nabla\mathbf{v} \tag{B.37}$$

The last term in the above can be calculated further:

$$\begin{aligned}\boldsymbol{\sigma}{:}\nabla\mathbf{v} &= \sigma_{\mathrm{pk}}\frac{\partial v_{\mathrm{p}}}{\partial x_{\mathrm{k}}} = \frac{1}{2}\sigma_{\mathrm{pk}}\frac{\partial v_{\mathrm{p}}}{\partial x_{\mathrm{k}}} + \frac{1}{2}\sigma_{\mathrm{kp}}\frac{\partial v_{\mathrm{k}}}{\partial x_{\mathrm{p}}} = \sigma_{\mathrm{pk}}\,\frac{1}{2}\left(\frac{\partial v_{\mathrm{p}}}{\partial x_{\mathrm{k}}} + \frac{\partial v_{\mathrm{k}}}{\partial x_{\mathrm{p}}}\right)\\ &= \boldsymbol{\sigma} : \left(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}}\right)/2 = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}\end{aligned} \tag{B.38}$$

The above manuver was possible since $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{T}}$ as mentiond in Section 2.3, and hence $\sigma_{\mathrm{pk}}\mathbf{i}_{\mathrm{p}}\mathbf{i}_{\mathrm{k}} = \sigma_{\mathrm{kp}}\mathbf{i}_{\mathrm{p}}\mathbf{i}_{\mathrm{k}}$ or $\sigma_{\mathrm{pk}} = \sigma_{\mathrm{kp}}$. Now, using Equations 2.23 and 2.24, the above expression can be transformed to the following:

$$\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} = (-p\,\mathbf{I} + 2\,\eta\,\dot{\boldsymbol{\varepsilon}}) : \dot{\boldsymbol{\varepsilon}} = \eta\,(2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}) = \eta\,\dot{\gamma}^2 \tag{B.39}$$

where $-p\,\mathbf{I} : \dot{\boldsymbol{\varepsilon}} = -p\,\delta_{ij}\dot{\varepsilon}_{ij} = -p\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}) = 0$, in accordance with Equation B.21. Using the above equation, Equation B.37 can be rewritten as shown below.

$$\rho\frac{du}{dt} = \rho\,\dot{h} - \nabla\cdot q + \eta\,\dot{\gamma}^2 \tag{B.40}$$

For incompressible liquids and solids, the change in internal energy is given by $du = c\,dT$ [49]. The term c [J/(kg · K)] is the specific heat, or heat capacity per unit mass. As such, the above equation can be presented as shown with Equation B.41.

$$\rho\,c\frac{dT}{dt} = \rho\,\dot{h} + \nabla\cdot(k\nabla T) + \eta\,\dot{\gamma}^2 \tag{B.41}$$

The temperature T is a macroscopic perception of random translational-, rotational- and vibrational motion for the smallest solid particles [12, 93]. The part of temperature only consisting of random translational motion, is the same motion as the random and the spontaneous part of the velocity $\mathbf{v}_\mathrm{I}$ (see Equation 2.2, Page 13), namely $[\mathbf{v}_\mathrm{I} - \mathbf{v}]$ for the same particles (see also Footnote 3, Page 378 and Footnote 6, Page 13). The addition in temperature dT/dt, to the already existing (room) temperature T, is due to the increase in random translational motion of the smallest[18] solid particles, when the CP is being deformed ($\dot{\boldsymbol{\varepsilon}} \neq 0$) by its surrounding force $\mathbf{f}$. This is described with the relationship between dT/dt and $\eta\,\dot{\gamma}^2 = \eta\,(2\,\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}})$ in Equation B.41. Of course, the increase in translational random motion will be, in part, converted to rotational- and vibrational motion, through collisions (and visa versa). With the above in mind, the energy Equation B.41 could be called the *equation of discontinuous random motion for the solid particles*. Quite the opposite, Equation B.25 represents the velocity part of $\mathbf{v}_\mathrm{I}$, which contributes to the overall smooth motion of the CP (see Equation 2.6, Page 13) and as such this equation could be called the *equation of smooth translational motion of the CP*.

B.8 Parallel Plate Viscometer

Because of convenience, the cylindrical coordinates will be used here. By using the general velocity field $\mathbf{v} = v_\mathrm{r}(r,\theta,z,t)\,\mathbf{i}_\mathrm{r} + v_\theta(r,\theta,z,t)\,\mathbf{i}_\theta + v_\mathrm{z}(r,\theta,z,t)\,\mathbf{i}_\mathrm{z}$ it is impossible to gain an analytical solution. But fortunately some reasonable assumptions about the flow can be made, which makes it easier to obtain such a solution:

1. With low Reynolds number (i.e. with low speed and high shear viscosity η) the flow is stable[19] and it is possible to assume a flow symmetry around the z-axis: $\mathbf{v} = v_\mathrm{r}(r,\theta,z,t)\,\mathbf{i}_\mathrm{r} + v_\theta(r,\theta,z,t)\,\mathbf{i}_\theta$.

2. Due to the circular geometry of the parallel plates and of the cylinder geometries that are involved (see for example Figure 10.23, Page 257), it is reasonable to

[18] The random motion of the larger solid particles cannot be understood as temperature. However, their random motion is always depleted by collisions of the surrounding smaller solid particles, which increases the random motion of the latter (the random motion of the former is maintained with the force $\mathbf{f}$). Now, the random motion of the smaller solid particles are again depleted by collisions of still smaller surrounding solid particles, and so forth. In this way, the random motion of the larger solid particles will eventually always contribute to the random (thermal) motion of the smallest solid particles and hence contributing to an increase in the CP temperature T. The increase in temperature is a reality first for the system of smallest solid particles. But this temperature (or heat) is also concomitantly conducted into the larger solid particles.

[19] See Footnote 8 on Page 56.

assume pure circular flow with θ-independence:

$$\mathbf{v} = v_\theta(r,z,t)\,\mathbf{i}_\theta \tag{B.42}$$

With the velocity field described with Equation B.42, then all the results of Chapter 7 are valid for the BTRHEOM as well for the ConTec viscometers.

The only possibility to make an analytical analysis of the BTRHEOM viscometer is to assume full slippage (i.e. a no adhesion case) at the outer and inner cylinder $r = R_\mathrm{o}$ and $r = R_\mathrm{i}$. Assuming this, then according to the Cauchy's stress principle, the mathematical condition which must apply there, consist of $\mathbf{n}\cdot\mathbf{T} = 0$, where $\mathbf{n}$ is the unit normal vector of the corresponding wall boundary (see the right illustration of Figure 10.22, Page 256), pointing into the bulk of test material. Using Equation 7.4 (Page 156) when using this condition at $r = R_\mathrm{o}$ and $r = R_\mathrm{i}$ produces the following:

$$-\mathbf{i}_\mathrm{r}\cdot\mathbf{T}|_{R_\mathrm{o}} = -\,\eta(\dot\gamma,t)\left[\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right]\Bigg|_{R_\mathrm{o}}\mathbf{i}_\theta = 0 \tag{B.43}$$

$$\mathbf{i}_\mathrm{r}\cdot\mathbf{T}|_{R_\mathrm{i}} = \eta(\dot\gamma,t)\left[\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r}\right]\Bigg|_{R_\mathrm{i}}\mathbf{i}_\theta = 0 \tag{B.44}$$

The above tells that the condition $\partial v_\theta(r,z,t)/\partial r - v_\theta(r,z,t)/r = 0$ applies at the outer and inner cylinder. Also, with $v_\theta = r\,\omega_\mathrm{t}$ at the top plate and $v_\theta = 0$ at the bottom plate, this same condition is also valid there. With this condition valid at all the wall boundaries, it is presumable to assume that this condition applies also everywhere else inside the test sample:

$$\frac{\partial v_\theta(r,z,t)}{\partial r} - \frac{v_\theta(r,z,t)}{r} = 0 \quad \forall \quad (r,z) \quad \in \quad \Omega\cup\partial\Omega \tag{B.45}$$

In other words, the condition $\partial v_\theta(r,z,t)/\partial r - v_\theta(r,z,t)/r = 0$ applies everywhere, whenever full slippage applies at the outer and inner cylinder. With this result, the shear rate, as described with Equation 7.3 (Page 156) is now given by Equation B.46.

$$\dot\gamma = \left|\frac{\partial v_\theta(r,z,t)}{\partial z}\right| = \frac{\partial v_\theta(r,z,t)}{\partial z} \geq 0 \tag{B.46}$$

According to Equation 7.7, the statement from Equation B.45 is equivalent to the condition $T_{\mathrm{r}\theta} = 0\ \forall\ (r,z) \in \Omega\cup\partial\Omega$. From Equation B.46 and with $T_{\mathrm{r}\theta} = 0$ and also assuming time independence for each discrete torque measurement, then Equation 7.6 is transformed to Equation B.47.

$$0 = \frac{\partial T_{z\theta}(r,z)}{\partial z} = \frac{\partial}{\partial z}\left(\eta(\dot\gamma)\,\frac{\partial v_\theta(r,z)}{\partial z}\right) = \frac{\partial}{\partial z}\left(\mu\,\frac{\partial v_\theta(r,z)}{\partial z} + \tau_\mathrm{o}\right) \tag{B.47}$$

In the above equation, the Bingham model is applied by using $\eta = \mu + \tau_\mathrm{o}/\dot\gamma$. With the boundary condition $v_\theta(r,0) = 0$ and $v_\theta(r,h) = r\omega_\mathrm{t}$, the solution[20] of Equation B.47 is as follows:

$$v_\theta(r,z) = \frac{r\,\omega_\mathrm{t}}{h}\,z \tag{B.48}$$

[20] $\mu\,\partial v_\theta/\partial z + \tau_\mathrm{o} = A(r) \Rightarrow \partial v_\theta/\partial z = (A(r)-\tau_\mathrm{o})/\mu \Rightarrow v_\theta = (A(r)-\tau_\mathrm{o})z/\mu + B(r);\ v_\theta(r,0) = 0 \wedge v_\theta(r,h) = r\omega_\mathrm{t} \Rightarrow B(r) = 0 \wedge (A(r)-\tau_\mathrm{o})/\mu = r\,\omega_\mathrm{t}/h.$

Combining Equations B.46 and B.48, gives the shear rate profile which applies for the full-slippage-case:

$$\dot{\gamma} = \frac{r\,\omega_\mathrm{t}}{h} \tag{B.49}$$

The shear stress applied **on** the top plate **from** the test material (c.f. Cauchy's stress principle) is $-\mathbf{i}_\mathrm{z} \cdot \mathbf{T}$. From Equations 7.4, B.45, B.46 and B.48, this term is equal to:

$$-\mathbf{i}_\mathrm{z} \cdot \mathbf{T} = -\eta\,\frac{\partial v_\theta(r,z)}{\partial z}\,\mathbf{i}_\theta = -\left(\mu\,\frac{r\,\omega_\mathrm{t}}{h} + \tau_\mathrm{o}\right)\,\mathbf{i}_\theta \tag{B.50}$$

The negative sign is because the concrete sample is always trying to slow down the rotating top plate. Now, the torque applied **from** the test material **on** the top plate is:

$$\begin{aligned}\hat{\mathbf{T}} &= -\hat{T}\,\mathbf{i}_\mathrm{z} = \int_{R_\mathrm{i}}^{R_\mathrm{o}}\int_0^{2\,\pi} r\,\mathbf{i}_\mathrm{r} \times (-\mathbf{i}_\mathrm{z} \cdot \mathbf{T}\,r\,d\theta\,dr) = \\ &= -\left[\frac{\pi\,\mu\,(R_\mathrm{o}^4 - R_\mathrm{i}^4)}{2\,h}\,\omega_\mathrm{t} + \frac{2\,\pi\,\tau_\mathrm{o}\,(R_\mathrm{o}^3 - R_\mathrm{i}^3)}{3}\right]\,\mathbf{i}_\mathrm{z} = -(H\,\omega_\mathrm{t} + G)\,\mathbf{i}_\mathrm{z}\end{aligned} \tag{B.51}$$

By plotting the measured torque $\hat{T}$ as a function of the angular velocities ω_t, one can connect these measured values with a straight line: $\hat{T} = H\,\omega_\mathrm{t} + G$. From its slope H and its point of intersection with the ordinate G, one can calculate the plastic viscosity μ and the yield value τ_o of the cement based material according to Equation B.52.

$$\mu = \frac{2\,h\,H}{\pi\,(R_\mathrm{o}^4 - R_\mathrm{i}^4)} \quad \wedge \quad \tau_\mathrm{o} = \frac{3\,G}{2\,\pi\,(R_\mathrm{o}^3 - R_\mathrm{i}^3)} \tag{B.52}$$

The above result is extracted directly from Equation B.51.

Appendix C

Additional Measurements

C.1 Compressive Strength of Concrete

Usually, for each concrete batch (see mix design of concrete in Table 4.3, Page 78), a number of 9 cubes ($100 \times 100 \times 100\,\text{mm}$) are casted and tested for compressive strength in accordance with Norwegian standards. The cubes are de-moulded at 24 hours, and three (of total of nine) cubes are tested at once, while the remaining six are stored in a curing tank at 20°C until testing at 7 and 28 days (in general). A single compressive strength value shown in Figures C.1 and C.2 is obtained as an average of three tested cubes. The cube strength is measured with the GALDABINI apparatus shown in the right illustration of Figure C.1. The left illustration of Figure C.1 demonstrates the measured compressive strength of concrete at $w/c = 0.4$. The left and right illustration of Figure C.2 shows the compressive strength at $w/c = 0.5$ and $w/c = 0.6$, respectively.

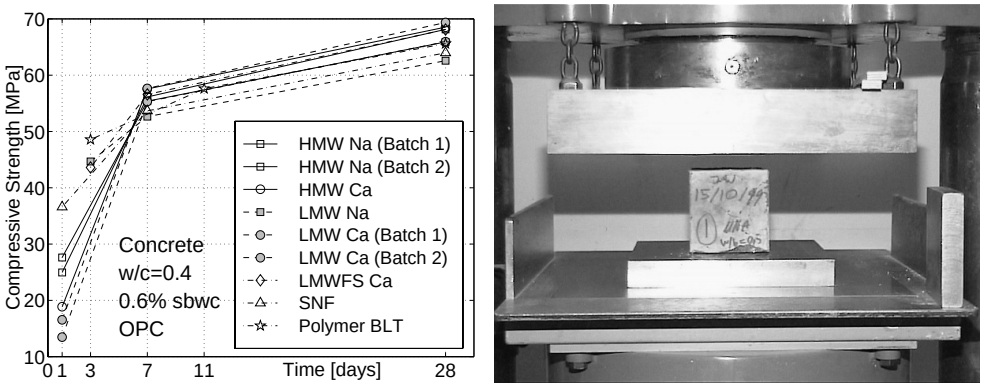


Figure C.1: To the left: Measured compressive strength of concrete at $w/c = 0.4$, using the different types of polymers. To the right: The measuring apparatus is the GALDABINI device.

C.2 First Setting Time of Mortar

The heat of hydration of mortar (see mix design of mortars in Tables 4.4 and 4.6, Page 78) is determined here by a semi-adiabatic calorimetry [116]. The semi-adiabatic

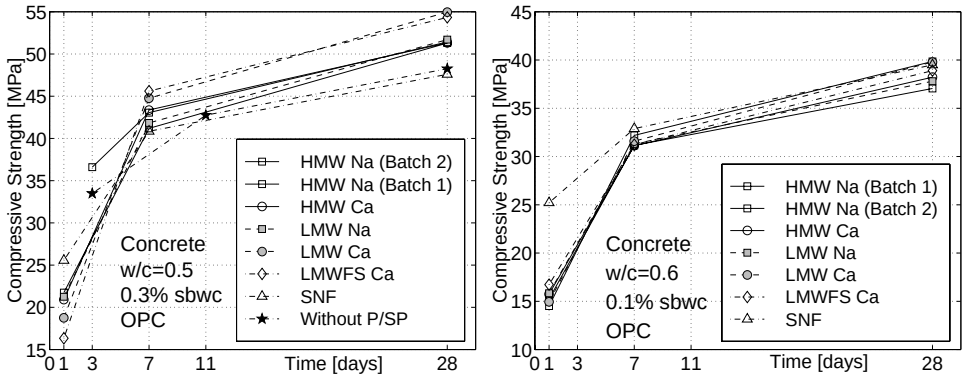


Figure C.2: Measured compressive strength of concrete at $w/c = 0.5$ (to the left) and at $w/c = 0.6$ (to the right), using the different types of polymers.

calorimeter is a small and closed insulated container made from 15 mm expanded polystyrene. Four liters of the mortar are placed inside this container. The ambient temperature T_o and the mortar temperature T are simultaneously recorded as a function of time t, by the use of thermocouples and a data logger. The heat convection from the container to the surroundings is low, but not negligible. Consequently the resulting measured temperature rise in the mortar sample do not correspond to pure adiabatic conditions. Error generated due to this is corrected with the help of Newton's law of heat convection $\mathbf{q}_\mathrm{N} = \hat{h}(T - T_\mathrm{o})\mathbf{n}$ (see Page 389 about this law).

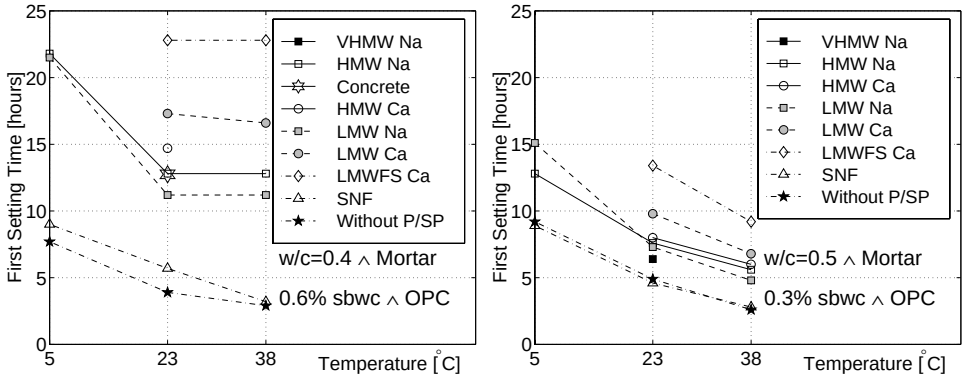


Figure C.3: The first setting time of mortars (OPC), using the different types of polymers. The only concrete batch that is mixed and measured, contains the **HMW Na** polymer.

The rate of increase in heat is calculated as $\dot{q} = \rho\,\dot{h} = \rho\,c\,dT/dt$, where the terms c and ρ are the specific heat $[\mathrm{J}/(\mathrm{kg}\cdot\mathrm{K})]$ and density $[\mathrm{kg}/\mathrm{m}^3]$ of the mortar sample ($dQ/dt = \iiint \dot{q}\,dV$, c.f. Equation B.30, Page 388). The accumulated heat development as a function of real time t is calculated with the following:

$$q_\mathrm{acc}(t) = \int_0^t \dot{q}(t')\,dt' \approx q_\mathrm{acc}(\mathrm{j}\Delta t) = \sum_{\mathrm{i}=1}^{\mathrm{j}} \rho\,c\,\Delta T_\mathrm{i} \tag{C.1}$$

The term ΔT_i is the change in temperature of the mortar (or concrete) during the time

interval Δt, at the real time $t = \mathrm{i}\Delta t$. Plotting $q_{\mathrm{acc}}(t)$ against maturity time $M(t)$ using the modified Arrhenius equation, results in a curve denoted as "isothermal heat development". Commonly used in Norway, the first setting time of mortar and concrete is determined with this type of curve: A $12.5\,\mathrm{kJ/kg_{cement}}$ heat increase compared to the underlying trend line during the period of setting, determines the first setting time in terms of maturity M (this is the so-called modified 12.5 kJ/kg-criteria [44]). This maturity time $M(t)$ can be calculated back to real time t. This is done in Figures C.3 and C.4. More precisely, these figures shows the first setting time (in real time t) of both OPC-mortars and FAC-mortars, with the different polymer types.

To indicate the relevance of the mortar results, one concrete batch is mixed and tested for comparison. This specific concrete batch contains the **HMW Na** polymer. The result of this is shown in the left illustration of Figure C.3. There it is shown a complete match between the mortar case and the concrete case. [As discussed in Section 4.3.2, the mix design of the OPC-mortar is calculated from the mix design of the concrete].

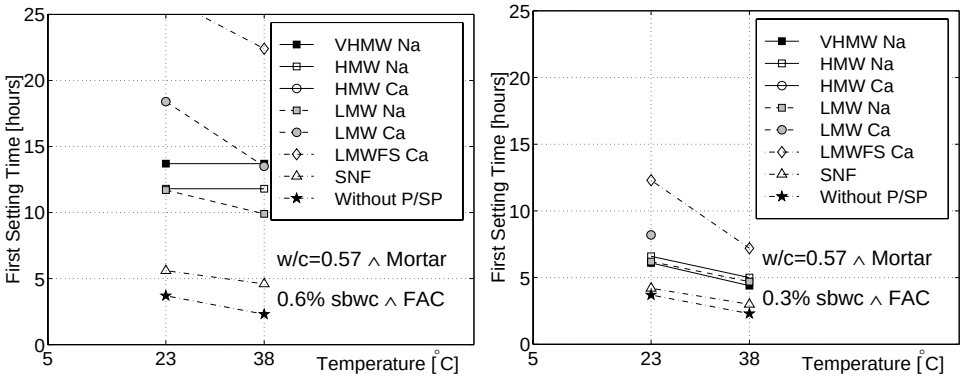


Figure C.4: The first setting time of mortars (FAC), using the different types of polymers.

Department of Structural Engineering
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
Telephone: +47 73 59 47 00; Fax: +47 73 59 47 01

"Reliability Analysis of Structural Systems using Nonlinear Finite Element Methods", C. A. Holm, 1990:23, ISBN 82-7119-178-0.

"Uniform Stratified Flow Interaction with a Submerged Horizontal Cylinder", Ø. Arntsen, 1990:32, ISBN 82-7119-188-8.

"Large Displacement Analysis of Flexible and Rigid Systems Considering Displacement-Dependent Loads and Nonlinear Constraints" K. M. Mathisen, 1990:33 ISBN 82-7119-189-6.

"Solid Mechanics and Material Models including Large Deformations", E. Levold, 1990:56, ISBN 82-7119-214-0, ISSN 0802-3271.

"Inelastic Deformation Capacity of Flexurally-Loaded Aluminium Alloy Structures", T. Welo, 1990:62, ISBN 82-7119-220-5, ISSN 0802-3271.

"Visualization of Results from Mechanical Engineering Analysis", K. Aamnes, 1990:63, ISBN 82-7119-221-3, ISSN 0802-3271.

"Object-Oriented Product Modeling for Structural Design", S. I. Dale, 1991:6, ISBN 82-7119-258-2, ISSN 0802-3271.

"Parallel Techniques for Solving Finite Element Problems on Transputer Networks", T. H. Hansen, 1991:19, ISBN 82-7119-273-6, ISSN 0802-3271.

"Statistical Description and Estimation of Ocean Drift Ice Environments", R. Korsnes, 1991:24, ISBN 82-7119-278-7, ISSN 0802-3271.

"Properties of concrete related to fatigue damage: with emphasis on high strength concrete", G. Petkovic, 1991:35, ISBN 82-7119-290-6, ISSN 0802-3271.

"Turbidity Current Modelling", B. Brørs, 1991:38, ISBN 82-7119-293-0, ISSN 0802-3271.

"Zero-Slump Concrete: Rheology, Degree of Compaction and Strength. Effects of Fillers as Part Cement-Replacement", C. Sørensen, 1992:8, ISBN 82-7119-357-0, ISSN 0802-3271.

"Nonlinear Analysis of Reinforced Concrete Structures Exposed to Transient Loading", K. V. Høiseth, 1992:15, ISBN 82-7119-364-3, ISSN 0802-3271.

"Finite Element Formulations and Solution Algorithms for Buckling and Collapse Analysis of Thin Shells", R. O. Bjærum, 1992:30, ISBN 82-7119-380-5, ISSN 0802-3271.

"Response Statistics of Nonlinear Dynamic Systems", J. M. Johnsen, 1992:42, ISBN 82-7119-393-7, ISSN 0802-3271.

"Digital Models in Engineering. A Study on why and how engineers build and operate digital models for decisison support", J. Høyte, 1992:75, ISBN 82-7119-429-1, ISSN 0802-3271.

"Sparse Solution of Finite Element Equations", A. C. Damhaug, 1992:76, ISBN 82-7119-430-5, ISSN 0802-3271.

"Some Aspects of Floating Ice Related to Sea Surface Operations in the Barents Sea", S. Løset, 1992:95, ISBN 82-7119-452-6, ISSN 0802-3271.

"Modelling of Cyclic Plasticity with Application to Steel and Aluminium Structures", O. S. Hopperstad, 1993:7, ISBN 82-7119-461-5, ISSN 0802-3271.

"The Free Formulation: Linear Theory and Extensions with Applications to Tetrahedral Elements with Rotational Freedoms", G. Skeie, 1993:17, ISBN 82-7119-472-0, ISSN 0802-3271.

"Høyfast betongs motstand mot piggdekkslitasje. Analyse av resultater fra prøving i Veisliter'n", T. Tveter, 1993:62, ISBN 82-7119-522-0, ISSN 0802-3271.

"A Nonlinear Finite Element Based on Free Formulation Theory for Analysis of Sandwich Structures", O. Aamlid, 1993:72, ISBN 82-7119-534-4, ISSN 0802-3271.

"The Effect of Curing Temperature and Silica Fume on Chloride Migration and Pore Structure of High Strength Concrete", C. J. Hauck, 1993:90, ISBN 82-7119-553-0, ISSN 0802-3271.

"Failure of Concrete under Compressive Strain Gradients", G. Markeset, 1993:110, ISBN 82-7119-575-1, ISSN 0802-3271.

"An experimental study of internal tidal amphidromes in Vestfjorden", J. H. Nilsen, 1994:39, ISBN 82-7119-640-5, ISSN 0802-3271.

"Structural analysis of oil wells with emphasis on conductor design", H. Larsen, 1994:46, ISBN 82-7119-648-0, ISSN 0802-3271.

"Adaptive methods for non-linear finite element analysis of shell structures", K. M. Okstad, 1994:66, ISBN 82-7119-670-7, ISSN 0802-3271.

"On constitutive modelling in nonlinear analysis of concrete structures", O. Fyrileiv, 1994:115, ISBN 82-7119-725-8, ISSN 0802-3271.

"Fluctuating wind load and response of a line-like engineering structure with emphasis on motion-induced wind forces", J. Bogunovic Jakobsen, 1995:62, ISBN 82-7119-809-2, ISSN 0802-3271.

"An experimental study of beam-columns subjected to combined torsion, bending and axial actions", A. Aalberg, 1995:66, ISBN 82-7119-813-0, ISSN 0802-3271.

"Scaling and cracking in unsealed freeze/thaw testing of Portland cement and silica fume concretes", S. Jacobsen, 1995:101, ISBN 82-7119-851-3, ISSN 0802-3271.

"Damping of water waves by submerged vegetation. A case study of laminaria hyperborea", A. M. Dubi, 1995:108, ISBN 82-7119-859-9, ISSN 0802-3271.

"The dynamics of a slope current in the Barents Sea", Sheng Li, 1995:109, ISBN 82-7119-860-2, ISSN 0802-3271.

"Modellering av delmaterialenes betydning for betongens konsistens", Ernst Mørtsell, 1996:12, ISBN 82-7119-894-7, ISSN 0802-3271.

"Bending of thin-walled aluminium extrusions", Birgit Søvik Opheim, 1996:60, ISBN 82-7119-947-1, ISSN 0802-3271.

"Material modelling of aluminium for crashworthiness analysis", Torodd Berstad, 1996:89, ISBN 82-7119-980-3, ISSN 0802-3271.

"Estimation of structural parameters from response measurements on submerged floating tunnels", Rolf Magne Larssen, 1996:119, ISBN 82-471-0014-2, ISSN 0802-3271.

"Numerical modelling of plain and reinforced concrete by damage mechanics", Mario A. Polanco-Loria, 1997:20, ISBN 82-471-0049-5, ISSN 0802-3271.

"Nonlinear random vibrations - numerical analysis by path integration methods", Vibeke Moe, 1997:26, ISBN 82-471-0056-8, ISSN 0802-3271.

"Numerical prediction of vortex-induced vibration by the finite element method", Joar Martin Dalheim, 1997:63, ISBN 82-471-0096-7, ISSN 0802-3271.

URN:NBN:no-3374

"Time domain calculations of buffeting response for wind sensitive structures", Ketil Aas-Jakobsen, 1997:148, ISBN 82-471-0189-0, ISSN 0802-3271.

"A numerical study of flow about fixed and flexibly mounted circular cylinders", Trond Stokka Meling, 1998:48, ISBN 82-471-0244-7, ISSN 0802-3271.

"Estimation of chloride penetration into concrete bridges in coastal areas", Per Egil Steen, 1998:89, ISBN 82-471-0290-0, ISSN 0802-3271.

"Stress-resultant material models for reinforced concrete plates and shells", Jan Arve Øverli, 1998:95, ISBN 82-471-0297-8, ISSN 0802-3271.

"Chloride binding in concrete. Effect of surrounding environment and concrete composition", Claus Kenneth Larsen, 1998:101, ISBN 82-471-0337-0, ISSN 0802-3271.

"Rotational capacity of aluminium alloy beams", Lars A. Moen, 1999:1, ISBN 82-471-0365-6, ISSN 0802-3271.

"Stretch Bending of Aluminium Extrusions", Arild H. Clausen, 1999:29, ISBN 82-471-0396-6, ISSN 0802-3271.

"Aluminium and Steel Beams under Concentrated Loading", Tore Tryland, 1999:30, ISBN 82-471-0397-4, ISSN 0802-3271.

"Engineering Models of Elastoplasticity and Fracture for Aluminium Alloys", Odd-Geir Lademo, 1999:39, ISBN 82-471-0406-7, ISSN 0802-3271.

"Kapasitet og duktilitet av dybelforbindelser i trekonstruksjoner", Jan Siem, 1999:46, ISBN 82-471-0414-8, ISSN 0802-3271.

"Etablering av distribuert ingeniørarbeid; Teknologiske og organisatoriske erfaringer fra en norsk ingeniørbedrift", Lars Line, 1999:52, ISBN 82-471-0420-2, ISSN 0802-3271.

"Estimation of Earthquake-Induced Response", Simon Ólafsson, 1999:73, ISBN 82-471-0443-1, ISSN 0802-3271.

"Coastal Concrete Bridges: Moisture State, Chloride Permeability and Aging Effects", Ragnhild Holen Relling, 1999:74, ISBN 82-471-0445-8, ISSN 0802-3271.

"Capacity Assessment of Titanium Pipes Subjected to Bending and External Pressure", Arve Bjørset, 1999:100, ISBN 82-471-0473-3, ISSN 0802-3271.

"Validation of Numerical Collapse Behaviour of Thin-Walled Corrugated Panels", Håvar Ilstad, 1999:101, ISBN 82-471-0474-1, ISSN 0802-3271.

"Strength and Ductility of Welded Structures in Aluminium Alloys", Miroslaw Matusiak, 1999:113, ISBN 82-471-0487-3, ISSN 0802-3271.

"Thermal Dilation and Autogenous Deformation as Driving Forces to Self-Induced Stresses in High Performance Concrete", Øyvind Bjøntegaard, 1999:121, ISBN 82-7984-002-8, ISSN 0802-3271.

"Some Aspects of Ski Base Sliding Friction and Ski Base Structure", Dag Anders Moldestad, 1999:137, ISBN 82-7984-019-2, ISSN 0802-3271.

"Electrode reactions and corrosion resistance for steel in mortar and concrete", Roy Antonsen, 2000:10, ISBN 82-7984-030-3, ISSN 0802-3271.

"Hydro-Physical Conditions in Kelp Forests and the Effect on Wave Damping and Dune Erosion. A case study on Laminaria Hyperborea", Stig Magnar Løvås, 2000:28, ISBN 82-7984-050-8, ISSN 0802-3271.

"Random Vibration and the Path Integral Method", Christian Skaug, 2000:39, ISBN 82-7984-061-3, ISSN 0802-3271.

URN:NBN:no-3374

"Buckling and geometrical nonlinear beam-type analyses of timber structures", Trond Even Eggen, 2000:56, ISBN 82-7984-081-8, ISSN 0802-3271.

"Structural Crashworthiness of Aluminium Foam-Based Components", Arve Grønsund Hanssen, 2000:76, ISBN 82-7984-102-4, ISSN 0809-103X.

"Measurements and simulations of the consolidation in first-year sea ice ridges, and some aspects of mechanical behaviour", Knut V. Høyland, 2000:94, ISBN 82-7984-121-0, ISSN 0809-103X.

"Kinematics in Regular and Irregular Waves based on a Lagrangian Formulation", Svein Helge Gjøsund, 2000-86, ISBN 82-7984-112-1, ISSN 0809-103X.

"Self-Induced Cracking Problems in Hardening Concrete Structures", Daniela Bosnjak, 2000-121, ISBN 82-7984-151-2, ISSN 0809-103X.

"Ballistic Penetration and Perforation of Steel Plates", Tore Børvik, 2000:124, ISBN 82-7984-154-7, ISSN 0809-103X.

"Freeze-Thaw resistance of Concrete. Effect of: Curing Conditions, Moisture Exchange and Materials", Terje Finnerup Rønning, 2001:14, ISBN 82-7984-165-2, ISSN 0809-103X

"Structural behaviour of post tensioned concrete structures". Flat slab. Slabs on ground", Steinar Trygstad, 2001:52, ISBN 82-471-5314-9, ISSN 0809-103X.

"Slipforming of Vertical Concrete Structures. Friction between concrete and slipform panel", Kjell Tore Fosså, 2001:61, ISBN 82-471-5325-4, ISSN 0809-103X.

"Some numerical methods for the simulation of laminar and turbulent incompressible flows", Jens Holmen, 2002:6, ISBN 82-471-5396-3, ISSN 0809-103X.

"Improved Fatigue Performance of Threaded Drillstring Connections by Cold Rolling", Steinar Kristoffersen, 2002:11, ISBN: 82-421-5402-1, ISSN 0809-103X.

"Deformations in Concrete Cantilever Bridges: Observations and Theoretical Modelling", Peter F. Takacs, 2002:23, ISBN 82-471-5415-3, ISSN 0809-103X.

"Stiffened aluminium plates subjected to impact loading", Hilde Giæver Hildrum, 2002:69, ISBN 82-471-5467-6, ISSN 0809-103X.

"Full- and model scale study of wind effects on a medium-rise building in a built up area", Jónas Thór Snæbjörnsson, 2002:95, ISBN82-471-5495-1, ISSN 0809-103X.

"Evaluation of Concepts for Loading of Hydrocarbons in Ice-infested water", Arnor Jensen, 2002:114, ISBN 82-417-5506-0, ISSN 0809-103X.

"Numerical and Physical Modelling of Oil Spreading in Broken Ice", Janne K. Økland Gjøsteen, 2002:130, ISBN 82-471-5523-0, ISSN 0809-103X.

"Diagnosis and protection of corroding steel in concrete", Franz Pruckner, 20002:140, ISBN 82-471-5555-4, ISSN 0809-103X.

"Tensile and Compressive Creep of Young Concrete: Testing and Modelling", Dawood Atrushi, 2003:17, ISBN 82-471-5565-6, ISSN 0809-103X.