ISBN 978-82-326-2564-2 (printed version)
ISBN 978-82-326-2565-9 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Ivar Eskerud Smith

Doctoral Thesis

Doctoral theses at NTNU, 2017:246

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Science and Technology
Department of Energy andProcess Engineering

NTNU

Doctoral theses at NTNU, 2017:246

Ivar Eskerud Smith

A
7-field Lagrangian slug capturing and slug tracking model with higher order methods

NTNU
Norwegian University of
Science and Technology

Ivar Eskerud Smith

A 7-field Lagrangian slug capturing and slug tracking model with higher order methods

Thesis for the degree of Philosophiae Doctor

Trondheim, August 2017

Norwegian University of Science and Technology
Faculty of Engineering
Science and Technology
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering
Science and Technology
Department of Energy and Process Engineering

ISBN 978-82-326-2564-2 (printed version)
ISBN 978-82-326-2565-9 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2017:246

MILJØMERKET 241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

Preface

This thesis is submitted for partial fulfilment of the requirements for the degree of Philosophiae Doctor (Ph.D.) at the Norwegian University of Science and Technology (NTNU).

The work has been performed at the Department of Energy and Process Engineering in the Faculty of Engineering Science and Technology. Professor Ole Jørgen Nydal has been the main supervisor, while Professor Ruud Henkes and Dr. Benjamin Sanderse from the Delft University of Science and Technology were co-supervisors.

The work has been financed by Shell, and was carried out from January 1st 2013 to December 31st 2016.

Abstract

In this thesis, a 7-field Lagrangian slug capturing and slug tracking model with higher order methods and an adaptive grid is investigated for predicting the behaviour of two-phase gas-liquid flow in multiphase flow pipelines. The model is capable of simulating both compressible and incompressible slugs, and pigs. The model has the possibility to simulate gas-liquid flow, including a liquid droplet field in the gas and entrained gas in the liquid. Walls consisting of multiple layers of different materials can be added to the pipes, and the energy equations for both the pipe walls and the fluids are solved. The mass, momentum and energy equations are solved in an iterative manner. Several additional tools and features have also been developed, like a steady state solver for the velocity, holdup, pressure and temperature, a unit-cell model which can be used as a standalone tool or as a sub-grid model in the dynamic model, fully period boundary conditions, curved pipe geometry, usage of tabulated PVT-files, and modelling of interfacial mass transfer. Higher order schemes are available for both spatial and temporal discretization. Different details in the two-fluid model have also been investigated, amongst other how to handle changes in the pipe cross-sectional area correctly for the border movement and level gradient. It is also shown how the upwind velocity must be modified by scaling factors to obtain the correct Bernoulli effect in the case of incompressible flow.

The work resulted in four papers. In Paper 1 the model was tested against large scale experimental data, and was shown to give good predictions of the slugging periods after including a liquid droplet field and including the separator in the simulations. In Paper 3 the slug capturing capabilities of the model are tested against experimental data from a medium scale flow loop, related to a project investigating cleaning of water distribution systems by use of slug flow. Paper 2 investigate the ability of higher order spatial and temporal schemes in detecting ill-posedness in the two-fluid model. Paper 4 is a continuation of Paper 2, and analyse the accuracy, stability and damping properties of different time integration schemes for the two-fluid model.

Acknowledgements

First of all, I would like to thank my supervisor Professor Ole Jørgen Nydal for giving me the opportunity of conducting this Ph.D. study. I have learned a lot from working in his research group, and the many discussions we have had has been much appreciated. I also want to thank him for several pleasant fishing trips with his boat and dinner gatherings at his house.

I am very grateful to Shell for their financial support of my Ph.D work. I would like to give a special thanks to my co-supervisors Professor Ruud Henkes and Dr. Benjamin Sanderse from Shell Technology Centre Amsterdam, for their support and valuable feedback during my Ph.D. I would also like to thank them for the multiple visits I had to Amsterdam to spend time with their research team to cooperate and exchange ideas, I have learned a lot from our cooperation.

A special thanks also to Dr. Jørn Kjølaas from SINTEF, for taking the time to answer my many questions and for the many interesting discussions we have had during my Ph.D. This has been much appreciated.

I also want to thank my Ph.D. colleagues Heiner Schumann, Andreas Akselsen, Tor Kjeldby, Andrea Schmueli, Mariana J. C. Diaz Arias, Joaquin Jose Vieiro Medina and Abraham Alejandro Parra Suarez from NTNU, and Maurice Hendrix from Delft University of Technology.

Furthermore I want to thank Jon Harald Kaspersen, my superior at SINTEF Petroleum, who allowed me to take a partial absence of leave in order to do this work.

Finally, I would like to thank my wife Hanne, for her patience and support during the course of this Ph.D.

Contents

Nomenclature

Abbreviations

CFL	Courant-Friedrichs-Lewy
GUI	Graphical User Interface
DWF	Downwind Weighting Factor
BDF	Backward Differentiation Formula
CN	Crank-Nicolson
FOU	First Order Upwind
TVD	Total Variation Diminishing

Roman letters

A	Cross sectional pipe area [m^2]	p	Pressure [Pa]
a_0, a_1, a_2	BDF coefficients	Q	Volumetric flow rate [m^3/s]
a_j^*	Control volume area to local area ratio [-]	Re	Reynolds number [-]
C	Droplet field concentration [kg/m^3]	R	Internal pipe radius [m]
C_l	Taylor bubble velocity distribution parameter [-]	R_b	Bend radius [m]
C_{slip}	Velocity slip constant [-]	R_s	Gas mass fraction at thermodynamic equilibrium [-]
C_D	Drag coefficient [-]	R_{cond}	Thermal resistivity [K/W]
D	Internal pipe diameter [m]	r, $\tilde{r}$	Ratio of consecutive gradients in flux limiter [-]
d_{32}	Sauter mean droplet diameter [m]	S	Pipe perimeter [m]
Fr	Froude number [-]	SF	Slug fraction [-]

g	Gravitational acceleration constant [m/s^2]	t, Δt	Time, time step [s]
Gr	Grashof number [-]	T	Temperature [K]
h_l	Liquid height [m]	u	Velocity [m/s]
h_k, h_f	Specific enthalpy [J/kg]	u_T	Terminal velocity [m/s]
h	Surface/contact conductivity [W/(m^{2}K)]	U_{slip}	Slip velocity [m/s]
k	Thermal conductivity [W/(mK]	V	Volume [m^3]
k_D	Deposition velocity [m/s]	W	Vertical or horizontal weighting factor [-]
L, Δx	Control volume length [m]	w	Momentum control volume extent factor [-]
M	Mass [kg]	We	Weber number [-]
m	Specific mass [kg/m^3]	x	Spatial coordinate [m]
Nu	Nusselt number [-]		
Pr	Prandtl number [-]		

Greek letters

α	Phase fraction [-]	μ	Dynamic viscosity [Pa$\cdot$s]
β_l	Wetted angle of the stratified liquid layer [rad]	Ψ	Flux limiter [-], Volume error correction [m^3/s]
∂	Derivative	ρ	Density [kg/m^3]
δ	Mean film thickness in annular flow [m]	σ	Surface tension [N/m]
ε	Hydraulic roughness [m]	τ	Shear stress [N/m^2]
ϕ	Entrainment/deposition rates [kg/s]	θ	Pipe inclination [°]
λ	Darcy friction factor [-]	Θ	Implicit fraction for the Crank-Nicolson method [-]
κ	Border velocity scaling constant [-]	ω_b	Volume time derivative scaling parameter [-]

Subscripts / superscripts

annu	Annular
amb	Ambient
b	Border / cell face
d	Deposition
e	Entrainment
f	Sub-field
g	Gas
gg	Gas in mixture gas
go	Gas in oil
gw	Gas in water
gl	Gas-liquid
h	Horizontal
hyd	Hydraulic
i	Interface
j	Control volume centre
J-1 / J	Left / right face of control volume
k	Mixture field
l	Liquid
m	Other mixture field
n	Time step
og	Oil in mixture gas
ol	Oil in mixture liquid
s	Sub control volume
src	Source/Sink
strat	Stratified
v	Vertical
wl	Water in mixture liquid
wg	Water in mixture gas

1 Introduction

In the petroleum industry, multiphase flow occurs when transporting oil and gas (and possibly water) in the same pipe through long multiphase pipeline systems. The behaviour of the flow can take many forms (flow patterns), depending on several parameters like fluid velocities, pipe diameter, pipe inclination, and the fluid properties. The fluid properties are again dependent on the pressure and temperature in the system, especially the gas density and the fluid viscosity. Certain flow patterns can cause significantly reduced production, or even such operational challenges that the pipeline must be abandoned. It is therefore of crucial importance to be able to predict the behaviour of the flow when investigating how to design the pipeline. The simplest of the flow patterns is stratified smooth flow, where the gas and liquid are separated by a clear and flat interface. Increasing the fluid velocities will increase the interfacial friction between the phases and result in waves forming on the interface. This flow pattern is referred to as stratified wavy flow. Depending on the conditions, these waves might become unstable, and grow large until they bridge the entire pipe cross-section and create a liquid plug. The gas pressure can then build up behind the liquid plug and accelerate it to high velocities through the pipe. This flow pattern is called hydrodynamic slug flow (sometimes just referred to as slug flow), and consists of alternating gas regions (Taylor bubbles) and liquid plugs (slugs). Such slugs results in increased wall friction and pressure loss, decreased production, and might even cause mechanical damage to the pipes. When a liquid blockage of the pipe cross-section occurs because of the pipe geometry, the flow is referred to as terrain slugging. Liquid then typically accumulates in low-points due to the pipe elevation. When liquid slugs accumulate at the bottom of a riser, the slug might grow depending on the gas velocity and upstream gas compressibility. When the pressure overcomes the hydrostatic pressure of the liquid in the riser, the gas will start to push the slug through the riser, and when the Taylor bubble enters the riser the slug is violently blown out. This type of slugging is called severe slugging, or riser-induced slugging. It is important to be able to predict the frequency and liquid volume of such severe slugs, especially for the design of slug catchers in receiving facilities so that they are not overfilled.

For a slug being pushed from the left to the right by the preceding Taylor bubble, the velocity of the left end of the slug (bubble nose) will be that of the nose of the Taylor bubble. The right end of the slug typically absorbs liquid (slug front) from the liquid below the upstream Taylor bubble, but this depends on several parameters like the difference in liquid velocity between the slug and the film below the Taylor bubble. It might happen that both ends of a slug are fronts absorbing liquid, or both ends might be Taylor-bubble noses. Empirical correlations to predict the velocity of a Taylor bubble nose has been extensively investigated; see for instance [1-6]. When simulating gas-liquid flow in petroleum pipelines, the so-called two-fluid model is used [7, 8]. This is a one-dimensional model, which is a result of averaging the three-dimensional Navier-Stokes equations since a full three-dimensional simulation of a pipeline would take an unreasonably long time. In the two-fluid model the slug velocity is a result of a combined effect of amongst others friction models and the numerical scheme. Previous work ([9]) has shown how the interfacial friction model in the commercial simulator LedaFlow ([10, 11]) had to be tuned at high holdup values to obtain slug velocities that corresponds with the aforementioned well-established slug velocity correlations.

This study is an extension of the work of Kjølaas and Kjeldby [12, 13], and presents a Lagrangian slug tracking model where the slug control volume velocity is set according to well established bubble nose velocity correlations. This automatically gives the desired velocities for the Taylor-bubbles, and enables sharp gas-liquid fronts at the slugs with no numerical diffusion. This allows for the use of a coarse grid for the slugs, giving fast simulations without the need for a sub-grid model like the unit-cell model concept ([14]) typically used by commercial simulators, like LedaFlow and OLGA [15]. However, one of the biggest challenges in the simulation of multiphase flow is how to initiate the slugs. Two different approaches can be used in the Sluggit model: Either initiate slugs on a coarse grid using mechanistic initiation criteria, or use a fine grid to resolve the natural growth of waves into slugs (so called slug capturing, first presented by Issa [16], and used by among others LedaFlow). Both approaches have their advantages and disadvantages. The initiation approach is fast, but is dependent of good correlations for when and how often to create a slug. The slug capturing approach on the other hand is significantly slower due to the fine grid and consequently reduced time step, but has the advantage that the natural growth of waves and slugs emerges from of solving the one-dimensional conservation equations for mass and momentum on a fine grid. The slug capturing approach however has the disadvantage that the two-fluid model can become ill-posed and yield grid-dependent solutions, which is still an unresolved issue.

In this Ph.D., a 7-field Lagrangian slug capturing and slug tracking model with higher order discretization methods has been investigated. In Paper 1 the model is compared with large scale experimental data. Furthermore, the effect of a droplet field and entrained gas in the slugs on the prediction of severe slugging was investigated. The effect of accounting for a finite size of the process facilities was also studied and shown to have a significant effect on the predicted slugging periods. In Paper 3, a combination of slug tracking and slug capturing has been tested and compared to experimental data. In Paper 2 and Paper 4, the effect of different higher order temporal and spatial methods on the prediction of the ill-posedness in the two-fluid model is investigated. The stability, accuracy and damping properties of different time integration methods are also studied, among others by a von Neumann analysis. Several details of the two-fluid model are also investigated in this thesis, among others:

- How to handle changes in the cross-sectional area in a two-fluid model with moving borders
- How to properly integrate the control volumes for a control volume than spans different pipes with different properties
- How to correctly obtain the correct Bernoulli effect when there are area changes
- How the level gradient term has to be modified when there are area changes

The model has several other capabilities, for some of them unfortunately there was not enough time left to use for publications. A unit-cell model that can also be used as a steady state solver is presented, used among others in Paper 2. The energy equations for the pipe walls and for a two-fluid model with moving borders are implemented, which together with the capabilities of periodic boundary conditions, compressible slugs, void in slug and interfacial mass transfer can be used to simulate heat exchange systems. This work will be continued by another Ph.D. student. The model also has the capability to use a bent pipe geometry, tabulated PVT-files, pigging, and several other features explained in Appendix Appendix C. A set of test cases that demonstrates and verifies some of the basic physics and functionality in the model are presented in Appendix Appendix B. The details of the Sluggit model are described in section 2.

2 Model description

The model is written in C++, using object oriented programming. Control volumes are represented by objects, like bubble section, slug section or pig section objects. Similarly, different types of border objects represent the borders (cell-faces) between the section objects (cell centres).

There are three different types of sections in the presented model: Bubble sections, slug sections and pig sections. There are two different concepts in the model for how the slugs are handled that requires different treatment in the equations: One where the slugs are treated as incompressible and one where the slugs are compressible. These are be referred to as the model with incompressible and compressible slugs respectively throughout the thesis. Pigs are treated as incompressible objects, similar to the incompressible slugs. As shown in the inheritance diagram in Figure 2, the incompressible slugs and pigs both inherit from a common incompressible plug base class, which in turn inherit from the general plug section class. The compressible slugs also inherit from the general plug section class, so that a plug section might be either compressible or incompressible.

The section and border objects are stored as linked lists, where one object has pointers to the previous and next objects in the list. In addition, consecutive sections of the same type belongs to the same *unit*. A unit contains pointers to the first and last section in the unit, and each section in the unit contain a pointer to the unit it belongs to. Bubble sections and compressible slugs are stored in bubble section units and compressible slug units respectively, while all types of incompressible plugs are stored in an incompressible plug unit. The main purpose of the unit class is that a consecutive list of incompressible objects will have the same velocity, and consequently the momentum equation is solved over the entire incompressible plug unit. The other types of units are only used as tools for traversing the lists of linked objects. The different sections and the unit concept is illustrated in Figure 1.

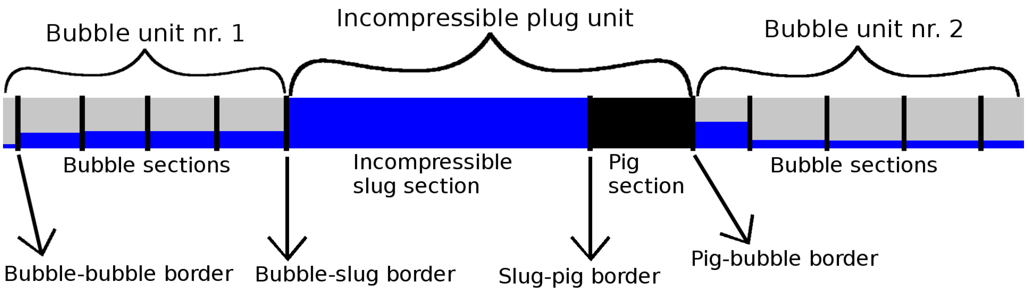


Figure 1: Illustration of the border, section and unit objects.

The mass, pressure, momentum and energy equations are described in section 2.9, but a short description of the momentum and pressure control volume classes will be presented here.
As shown in Figure 2 the unit class inherits from the momentum control volume class, which means that it has the possibility to be a part of the momentum equation system. The same applies for the border and section objects, which also inherit from the momentum control volume class. Note that momentum is not solved for the sections even though the section class inherits from the momentum control volume class. Similarly, the pressure equation is not solved for the border objects even though they inherit from the pressure control volume class, where the pressure variable is stored. This is done amongst others to have access to the velocity and pressure variables for these objects, but these variables are then calculated for instance as averages from the neighbouring objects where the respective variables are defined and solved. Another reason for the presented inheritance hierarchy is that one has the possibility to easily test other types of grids. Depending on the grid, each object that inherits from the momentum or pressure control volume classes must implement a function that specifies if momentum and pressure is solved or not. If momentum or pressure is not solved, a function must be implemented that calculates these variables from neighbouring objects where the variables are defined. In addition, a function must be implemented that returns a pointer to the next or previous momentum or pressure control volume. These functions are used to traverse the list of objects for which momentum or pressure are solved. Currently, only a staggered grid is implemented. The momentum and pressure control volume classes in turn inherit from a common control volume base class. This is done as there are some common variables for all control volumes regardless of type, like mass, length, and the absolute position at the start and end of the control volume.

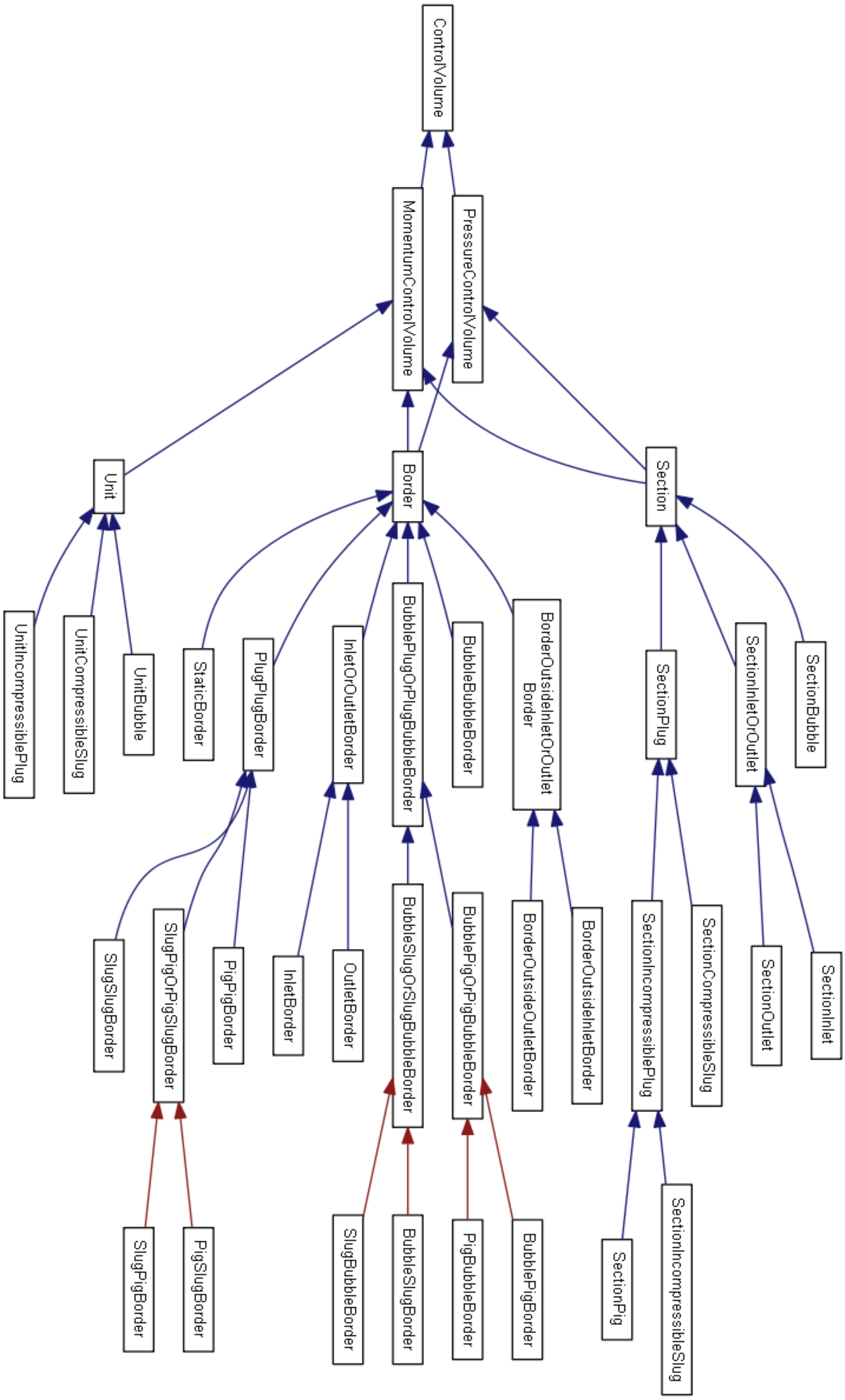


Figure 2: Inheritance diagram for the border, section and unit classes.

2.1 Phase subscripts and mixture formulations

The Sluggit code has 7 fields implemented. The fields are:
Gas bubbles in oil (*go*), gas bubbles in water (*gw*), continuous oil in mixture liquid (*ol*), continuous water in mixture liquid (*wl*), continuous gas in mixture gas (*gg*), oil droplets in mixture gas (*og*) and water droplets in mixture gas (*wg*). The energy and momentum equations are solved for the mixture liquid (*l*) and mixture gas (*g*) fields, while the velocities for the sub-fields are calculated from slip relations. A sub-field is defined as a field that is not a mixture field (the 7 aforementioned fields). The mass equations are solved for all sub-fields. The mixture relations are as follows:

Mixture liquid:

$$\begin{aligned} \alpha_l &= \alpha_{go} + \alpha_{gw} + \alpha_{ol} + \alpha_{wl} \\ \rho_l &= \frac{\rho_{go}\alpha_{go} + \rho_{gw}\alpha_{gw} + \rho_{ol}\alpha_{ol} + \rho_{wl}\alpha_{wl}}{\alpha_l} \end{aligned} \tag{1}$$

Mixture gas:

$$\begin{aligned} \alpha_g &= \alpha_{gg} + \alpha_{og} + \alpha_{wg} \\ \rho_g &= \frac{\rho_{gg}\alpha_{gg} + \rho_{og}\alpha_{og} + \rho_{wg}\alpha_{wg}}{\alpha_g} \end{aligned} \tag{2}$$

Mixture fields are referred to by the subscript *k*, while sub-fields are referred to by *f*. Three-phase flow or oil-water flow has not been in the scope of this thesis, and consequently the oil in water and water in oil fields are missing.

2.2 Grid

The employed grid is a staggered grid, where the masses, densities and pressure are stored in the section centres, while the velocities are stored at the borders. The mass and pressure equations are solved for the sections, while momentum is solved at the borders. For the incompressible plugs however, momentum is solved over the entire incompressible plug unit (non-staggered).

2.3 Indexing convention

The time index for the current and next time step is represented by n and n+1 respectively, and is shown as an upper index with the variables. Variables defined between time n and n+1 are referred to by $n+\frac{1}{2}$, for instance the volume covered by a border as it moves from position x^n to x^{n+1}. Subscript j denotes values defined at the control volume centre, while subscripts J-1 and J denote values at the left and right face of the control volume. Subscripts j-1 and j+1 represents the values at the previous and next control volumes.
The indexing convention for the mass, pressure and energy equations is shown in Figure 3.

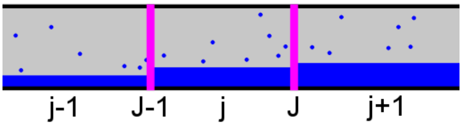


Figure 3: Indexing convention for the mass, pressure and energy equations.

The momentum indexing convention for bubble-bubble borders and incompressible plugs are shown in Figure 4 and Figure 5.

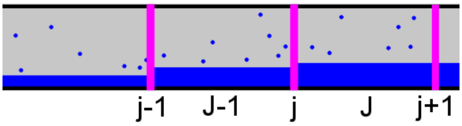


Figure 4: Indexing convention for the momentum equation for bubble-bubble borders.

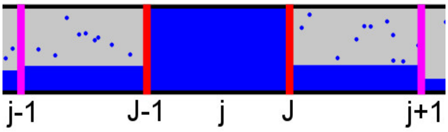


Figure 5: Indexing convention for the momentum equation for incompressible plugs.

For compressible slugs the momentum is solved at the slug borders, but the momentum control volume is modified compared to the bubble-bubble borders. To avoid having any continuous gas in the momentum control volume for the compressible slugs, the extent of the left part of the control volume for a bubble-slug border is set to zero (the standard being that the control volume extends half way into each neighbouring section). This means that the left end and centre of the control volume is the same location, as shown in Figure 6. This also means that the velocity at the left end-point is no longer undefined, and there is no need to use a convection scheme at this location. Similarly, the extent of the right part of the control volume for a slug-bubble border is also set to zero. As will be described further below by equation (4), the mass residing in this control volume is then no longer simply the average of the neighbouring section masses.

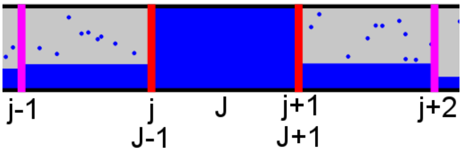


Figure 6: Indexing convention for the momentum equation for compressible slugs.

Since the control volume for the plugs does not extend into the neighbouring bubble sections, there will be a part of these bubble sections that is not covered by any momentum control volume. If the control volume for instance for a bubble-bubble border to the right of a slug-bubble border extends only half way into the bubble section to the left, the left half of the bubble section (to the right of the slug) would not be included in any momentum equation. The momentum leaving one control volume would then not be equal to the momentum entering the next, and momentum would not be conserved. One could possibly also miss momentum from mass sources if they are located at a location not covered by any control volume. Consequently, the control volume for a bubble-bubble border next to a plug border is extended all the way to the plug, as illustrated in Figure 5 and Figure 6.

Note that for the regular bubble-bubble border control volumes, the extent into the neighbouring sections is not necessarily equal to half of the length of the section, but rather the length that gives half of the section volume. For a section with a constant cross-sectional area this will be identical to half the section length, but not for a section spanning pipes with different diameters as illustrated in Figure 7. This is also described in more detail in the grid management section (section 2.15).

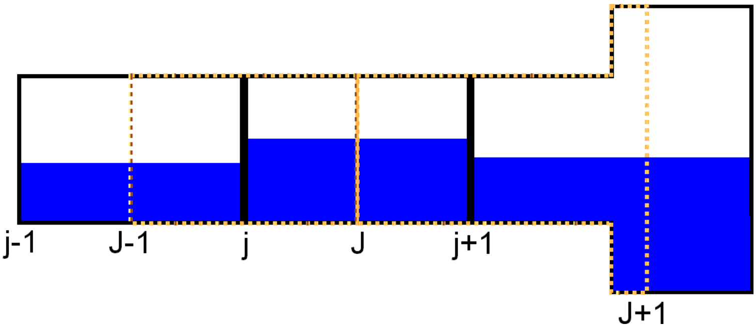


Figure 7: Illustration of a section spanning multiple pipes. Section *J*+1 spans two pipes with different diameters (but equal lengths). The right part of the border control volume for the border at index *j*+1 will extend a length that corresponds to half of the volume of section *J*+1.

The reason for choosing the control volume length of the borders based on volume is to simplify the calculation of properties for the border control volume that are not defined within the border control volume. For instance, the mass in a border control volume is calculated as:

$$M_j = 0.5M_{J-1} + 0.5M_J \tag{3}$$

This will only be the case if the left and right half of the control volume spans half of the left and right section volumes.

Properties in a border control volume calculated from section values, for instance mass, can then be calculated as follows:

$$M_j = w_{J-1}M_{J-1} + w_J M_J \tag{4}$$

Here w_{J-1} and w_J are weighting factors representing the fraction of the section volume covered of the left and right section. For instance, if the left part of the border control volume extends the entire left section, w_{J-1} equals one, while if the left part has zero extent like for a compressible slug border, w_{J-1} equals zero.

2.4 CFL criterion

The time step in the simulations can either be constant, or calculated from the Courant-Friedrichs-Lewy (CFL) condition [17], which is the default option:

$$CFL = \frac{u_{\max}\Delta t}{\Delta x} \leq 1 \tag{5}$$

Here $u_{\max}$ is the maximum velocity of all sub-fields.
Equation (5) can also be written as:

$$u_{\max}\Delta t \leq \Delta x \tag{6}$$

This means that the fluid in any cell should not move further than the length of the grid cell. If not, the fluid in one cell could move past the next cell during one time step, so that the next cell would never experience/feel that fluid moving by at all. It is possible to use larger time steps than this, but this would result in more diffusion and a less accurate result. The larger the CFL number, the closer an answer to a steady state like situation one will get. The CFL-criterion must be evaluated in all of the sections in the system, and the minimum time step obtained must be used.
Taking into account that the borders might move, the following criterion (using the indexing convention for the bubble-bubble borders):

$$\left(x_{j-1} + u_{b,j-1}\Delta t\right) \leq \left(x_j + u_{f,j}\Delta t\right) \leq \left(x_{j+1} + u_{b,j+1}\Delta t\right) \tag{7}$$

This can be written as two separate criteria:

$$\begin{aligned} \Delta t &\leq \frac{x_j - x_{j-1}}{\left(u_{b,j-1} - u_{k,j}\right)} \\ \Delta t &\leq \frac{x_{j+1} - x_j}{\left(u_{k,j} - u_{b,j+1}\right)} \end{aligned} \tag{8}$$

The two criteria in equation (8) involves three borders (two sections). For a single section, the following criteria applies:

$$\begin{aligned}\Delta t &\le \frac{x_{j+1}-x_j}{\left(u_{b,j}-u_{k,j+1}\right)}\\ \Delta t &\le \frac{x_{j+1}-x_j}{\left(u_{k,j}-u_{b,j+1}\right)}\end{aligned} \tag{9}$$

These two criteria describe that a lump of fluid located near a border should not move past the new positions of the neighbouring borders within one time step. Note that these criteria should only be included if the result is positive. A negative result simply means that the border is moving away faster than the fluid, so that a lump of fluid will never be able to move past the neighbouring border regardless of the time step (a negative time step would be required, which is unphysical). One can also see from equation (9) that when the fluid velocity and border velocity approaches the same value, the time step can be arbitrarily large.

The minimum time step calculated from equation (9) for all sub-field velocities will then be the limiting time step. Finally, the time step found from equation (9) is multiplied by a user-specified constant, which is the applied CFL number.

2.5 Volume fraction error

After solving the mass, momentum and energy equations, the phase fractions (holdups) are computed by dividing the specific masses by the densities (found from the associated equation of state with the new temperature and pressure):

$$\alpha_f = \frac{M_f}{\rho_f V} \tag{10}$$

When doing so, one will find that the sum of the phase fractions in a cell might not sum exactly to 1:

$$\sum_f \frac{M_f}{\rho_f V} \neq 1 \tag{11}$$

We define the volume calculated from the masses and densities as:

$$V_\rho = \sum_f \frac{M_f}{\rho_f} \tag{12}$$

We define the volume error, which is the difference between the total volume calculated from the masses and densities subtracted by the physical volume.

$$\Delta V_{err} = V_\rho - V \tag{13}$$

We also define a relative error called the volume fraction error by dividing the volume error by the physical volume:

$$V_{err} = \frac{\Delta V_{err}}{V} \tag{14}$$

This reflects the fact that there is a discrepancy between the fluid densities (found from the pressure and temperature) and the fluid masses (found from the mass equations). We can try to correct for this problem by adding a source term in the pressure equation in the next time step:

$$\Psi^{n+1} = \frac{\Delta V_{err}^n}{\Delta t^{n+1}} \tag{15}$$

This term ensures that mass is conserved over time, and that the sum of the phase fractions stays close to one.

Note that the volume fraction error is calculated by summing all sub-fields, and not only the mixtures fields.

2.6 Spatial convection schemes

Sometimes a variable is needed at a location where it is not defined, for instance the velocity at the section centre, or density or holdup at the borders. Advanced methods exists that takes into account the characteristic speeds the information in the system will travel with, both the pressure waves and fluid waves. This requires the use of a Riemann-solver, for instance a Roe-solver [18] which recognizes shock waves and transports all characteristics nicely. This is however a much more cumbersome approach and requires the construction of a matrix at each cell-face. Instead, the commonly used flux-limiter (convection scheme) approach is used in the Sluggit code. Multiple flux-limiters are implemented, amongst others the first order upwind scheme (FOU) [19] and the formally third order upwind NOTABLE scheme [20]. The remaining limiters are described further below in Table 1.

Any variable that is calculated from a convection scheme is shown with a "hat" above, for instance upwind velocity at a section centre ($\hat{u}$) or the upwind specific mass at the borders ($\hat{m}$). The calculation of a variable with the first order upwind scheme can be performed by the use of the upwind coefficient X , which is defined such that it has the value 1 if the information at the location travels from left to right, and zero if the information travels from right to left.

This can be calculated as follows:

$$X_{k,j}^{n+1} = 0.5\left(1 + sign\left(u_{k,j}^{n+1} - u_{k,b}^{n+1}\right)\right) \tag{16}$$

The *sign* function returns the sign of the input. Note that in the current code the *sign* function is designed to return zero if the input is zero, which means that the upwind coefficient will get a value of 0.5 if $u_{k,j}^{n+1} == u_{k,b}^{n+1}$. This will give symmetric solutions in the rare cases of exactly equal velocities.

A general variable Φ can then be calculated at a location J where it is not defined as follows (using the indexing convention for the mass equation):

$$\hat{\Phi}_{J,FOU}^{n+1} = X_J^{n+1}\Phi_j^{n+1} + \left(1 - X_J^{n+1}\right)\Phi_{j+1}^{n+1} \tag{17}$$

The NOTABLE scheme is implemented similarly as the FOU scheme, and consists of a first order upwind part using new values plus a third order term using values from time step *n*:

$$\hat{\Phi}_{J,NOTABLE}^{n+1} = \hat{\Phi}_{J,FOU}^{n+1} + DWF_J^n\left(\tilde{r}_J\right)\left(\Phi_{j+1}^n - \Phi_j^n\right) \tag{18}$$

In equation (18) and in the following explanations of the NOTABLE scheme the direction of the flow is assumed from left to right. The variable *DWF* is called the Downwind Weighting Factor, and is defined by:

$$\begin{aligned} DWF_J\left(\tilde{r}_J\right) &= \frac{\varphi_J - \tilde{r}_J}{1 - \tilde{r}_J} \\ \varphi_J &= c_1\tilde{r}_J^3 + c_2\tilde{r}_J^2 + c_3\tilde{r}_J \\ \tilde{r}_J &= \frac{\Phi_j^n - \Phi_{j-1}^n}{\Phi_{j+1}^n - \Phi_{j-1}^n} \end{aligned} \tag{19}$$

If the grid size is not uniform, the coefficients c_1, c_2 and c_3 in the NOTABLE scheme are functions of the ratios of the grid sizes:

$$\begin{aligned} r_1 &= \frac{\Delta x_{j+1}}{\Delta x_j} \\ r_2 &= \frac{\Delta x_j}{\Delta x_{j-1}} \\ \Upsilon &= \left(r_1 + 1\right)^2\left(r_2 + 1\right) \\ c_1 &= \frac{r_2^2\left(r_1^2 + 4r_1 + 4\right) + 2r_1r_2 + 4r_2 + 1}{\Upsilon\left(r_2 + 1\right)} \\ c_2 &= -\frac{\left(2r_2 + r_1r_2 + 1\right)\left(r_1 + 4r_2 + 2r_1r_2 + 3\right)}{\Upsilon(r_2 + 1)} \\ c_3 &= \frac{r_1^2\left(2r_2 + 1\right) + 3r_1 + 5r_2 + 6r_1r_2 + 3}{\Upsilon} \end{aligned} \tag{20}$$

If the grid ratios r_1 and r_2 equals 1 (uniform grid), the coefficients c_1, c_2 and c_3 from equation (20) becomes 1, -2.5 and 2.5 respectively.

The Downwind Weighting Factor is similar to the more commonly used *flux limiters*, and are related as shown in equation (24). If the *DWF* parameter is zero the first order upwind scheme is recovered, while a value of one gives the downwind value. If the value of *DWF* is outside the interval 0 – 1, the first order upwind scheme is used.

Note that the ratio $\tilde{r}$ in equation (19) for the NOTABLE scheme is not the same ratio as is typically used in flux limiters (which is the ratio of the consecutive gradients.), but is the same ratio as defined by Leonard [21], which introduced the Normalized Variable Formulation (NVF). Gaskell and Lau [22] in addition introduced the Convection Boundedness Criterion (CBC), specifying the necessary conditions for a limiter to give monotone solutions.

A general approach for using a higher order flux limiter Ψ can be expressed by the first order upwind value, plus an anti-diffusive term using values from the previous time step (here the flow is assumed to go from left to right):

$$\hat{\Phi}_J^{n+1} = \hat{\Phi}_{J,FOU}^{n+1} + \frac{1}{2}\Psi_J^n\left(r_J^n\right)\left(\Phi_j^n - \Phi_{j-1}^n\right) \tag{21}$$

The ratio r_J^n is the ratio of the consecutive gradients, first introduced by Sweby [23]:

$$r_J^n = \frac{\dfrac{\Phi_{j+1}^n - \Phi_j^n}{\Delta x_{j,j+1}}}{\dfrac{\Phi_j^n - \Phi_{j-1}^n}{\Delta x_{j-1,j}}} \tag{22}$$

Normally one assumes equal grid spacing, and equation (22) simplifies and can be related to the ratio $\tilde{r}$:

$$r_J^n = \frac{\Phi_{j+1}^n - \Phi_j^n}{\Phi_j^n - \Phi_{j-1}^n} = \frac{1-\tilde{r}_J^n}{\tilde{r}_J^n}$$

$$\text{or} \qquad (23)$$

$$\tilde{r}_J^n = \frac{1}{r_J^n + 1}$$

Furthermore, one has the following relation between the DWF factor and the flux limiter Ψ:

$$\begin{aligned} DWF_J^n\left(\Phi_{j+1}^n - \Phi_j^n\right) &= \frac{1}{2}\Psi_J^n\left(\Phi_j^n - \Phi_{j-1}^n\right) \\ \Psi_J^n = 2DWF_J^n\frac{\left(\Phi_{j+1}^n - \Phi_j^n\right)}{\left(\Phi_j^n - \Phi_{j-1}^n\right)} &= 2r_J DWF_J^n \end{aligned} \qquad (24)$$

The most commonly used limiters are the so-called Total Variation Diminishing (TVD) limiters [24], which ensures to preserve monotonicity and does not create unphysical oscillations. That is, if Φ^n monotonically increasing (or decreasing) in space, then so is Φ^{n+1}.

$$\begin{aligned} 0 \le \Psi(r) \le \min(2r, 2) \quad & r \ge 0 \\ \Psi(r) = 0 \quad & r \le 0 \end{aligned} \qquad (25)$$

From equation (25) we can see that the first order upwind scheme is actually a TVD scheme ($\Psi = 0$). For a TVD limiter to be second order accurate, the following conditions must be satisfied:

$$\begin{aligned} r \le \Psi(r) \le 1 \quad & 0 \le r < 1 \\ 1 \le \Psi(r) \le r \quad & r \ge 1 \end{aligned} \qquad (26)$$

TVD schemes that in addition have a slope of 0.75 at $r = 1$ are formally third order accurate (on a uniform grid). Some limiters are also said to be symmetric, if they fulfil the condition $r\Psi\left(\frac{1}{r}\right) = \Psi(r)$. Physically this means that increasing and decreasing gradients are treated equally. It is important that r is limited to be larger than or equal to zero, as a negative sign means a local change of sign of the consecutive gradients, and we revert to the first order upwind scheme. If the value of r is one, the gradients are equal and we have a linear slope.

Several limiters can be chosen in the Sluggit framework, listed in Table 1. Note that some limiters use r, and some use $\tilde{r}$. The limiters have mainly been found in the following places: [23, 25-30]. Not all of these limiters fulfil the criteria to be classified as TVD, and not all are symmetric. A symmetric limiter is indicated with capital letter S in the comments in Table 1. In Figure 8 - Figure 12 the limiter functions are plotted together with the second order TVD region. Note that the actual order of accuracy of the presented limiters will not necessarily be the same as the formal accuracy listed in Table 1, but will also depend on the truncation error.

Table 1: Flux limiters implemented in the Sluggit program.

Limiter	Ψ	Id	Order	Comment
FOU	0	0	1	TVD (First order upwind)
NOTABLE	Same as IsNaS, but the formulation in equations (18)-(19) will be used instead.	1	3	TVD (Same as IsNaS)
Central	$\min(r, 2.0)$	3	2	
IsNaS	$\frac{(r+\|r\|)(1.5r+0.5)}{(1+r)^2}$	4	3	TVD (Identical to NOTABLE)
Van Albada	$\frac{r^2+r}{r^2+1}$	2	2	TVD, S
Ospre	$\frac{1.5r(r+1)}{r^2+r+1}$	5	2	TVD, S
Van Leer	$\frac{r+\|r\|}{r+1}$	12	2	TVD, S
Superbee	$\max(0, \max(\min(2r,1), \min(r,2)))$	8	2	TVD, S
MinMod	$\max(0, \min(r,1))$	6	2	TVD, S

Monotonized Central	$\max\left(0,\min\left(\min\left(2r,0.5+0.5r\right),2\right)\right)$	14	2	TVD, S
Sweby	$\max\left(0,\max\left(\min\left(\beta_{sweby}r,1\right),\min\left(r,\beta_{sweby}\right)\right)\right)$	7	2	TVD, S
Osher	$\max\left(0,\min\left(r,\beta_{osher}\right)\right)$	18	2	TVD
Umist	$a=\min\left(2r,0.25+0.75r\right)$ $b=\min\left(0.75+0.25r,2\right)$ $\max\left(0,\min\left(a,b\right)\right)$	13	2	TVD, S
Koren	$\max\left(0,\min\left(\min\left(2r,\frac{2+r}{3}\right),2\right)\right)$	16	2	TVD
Smart	$\max\left(0,\min\left(2r,\min\left(0.75r+0.25,4\right)\right)\right)$	10	3	
H_QUICK	$\frac{2\left(r+\lvert r\rvert\right)}{r+3}$	9	3	TVD
Charm	$\frac{r\left(1+3r\right)}{\left(1+r\right)^2}$	11	3	TVD
HCUS	$\frac{1.5\left(r+\lvert r\rvert\right)}{r+2}$	15	2	TVD
Van Albada2	$\frac{2r}{1+r^2}$	17	2	
SOU	0.5	19	2	Second order upwind
Lax Wendroff	1.0	20	2	
Fromm	$0.5r+0.5$	21	2	S
VONOS	$\frac{2}{\tilde{r}}\left(10\tilde{r}-\tilde{r}\right) \quad 0\le\tilde{r}<\frac{3}{74}$ $\frac{2}{\tilde{r}}\left(\frac{3}{8}\left(1+2\tilde{r}\right)-\tilde{r}\right) \quad \frac{3}{74}\le\tilde{r}<0.5$ $\frac{2}{\tilde{r}}\left(1.5\tilde{r}-\tilde{r}\right)=1 \quad 0.5\le\tilde{r}<\frac{2}{3}$ $\frac{2}{\tilde{r}}\left(1-\tilde{r}\right) \quad \frac{2}{3}\le\tilde{r}\le1$	22	2-3	

QUICK	$\frac{2}{\tilde{r}}\left(\frac{3}{8}(1+2\tilde{r})-\tilde{r}\right)$		23	3	
SMART	$\frac{2}{\tilde{r}}(10\tilde{r}-\tilde{r})$	$0 \le \tilde{r} < \frac{3}{74}$	24	2	Bounded QUICK
	$\frac{2}{\tilde{r}}\left(\frac{3}{8}(1+2\tilde{r})-\tilde{r}\right)$	$\frac{3}{74} \le \tilde{r} < \frac{5}{6}$			
	1	$\frac{5}{6} \le \tilde{r} \le 1$			
ACUTE	$2\frac{(1-\tilde{r})}{\tilde{r}}(1-(1-\tilde{r})(1-8\tilde{r}(\tilde{r}-0.5)(\tilde{r}-0.75)$		25	3	
ACUTER	$2\frac{(1-\tilde{r})}{\tilde{r}}\left(-\frac{19}{4}+\left(\frac{71}{4}+(-28+15\tilde{r})\tilde{r}\right)\tilde{r}\right)\tilde{r}$		26	3	

Figure 8 - Figure 12 shows the limiter functions together with the second order TVD region.

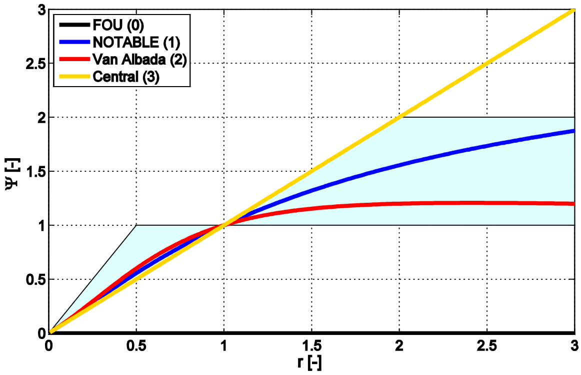


Figure 8: TVD region and limiter function for limiters with id 0-3.

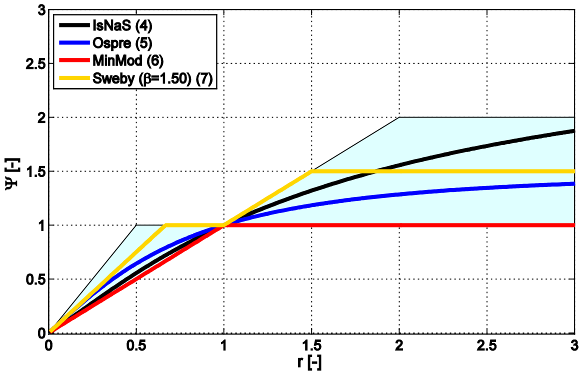


Figure 9: TVD region and limiter function for limiters with id 4-7.

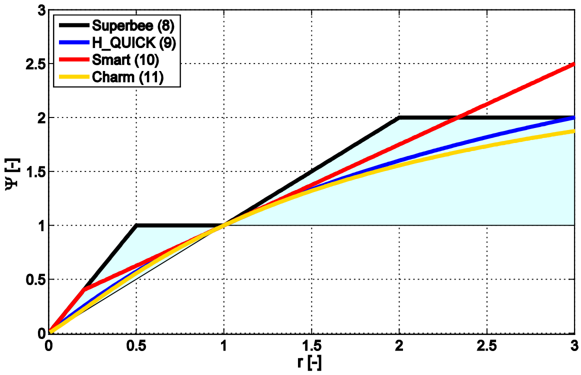


Figure 10: TVD region and limiter function for limiters with id 8-11.

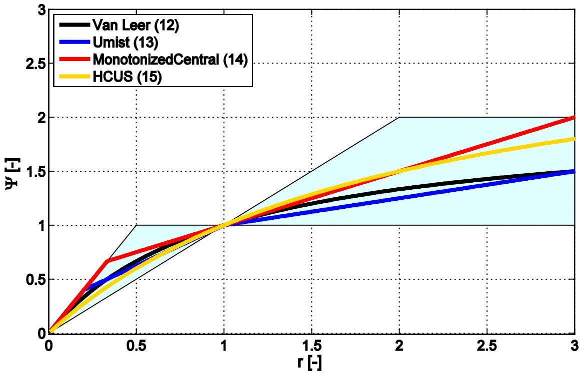


Figure 11: TVD region and limiter function for limiters with id 12-15.

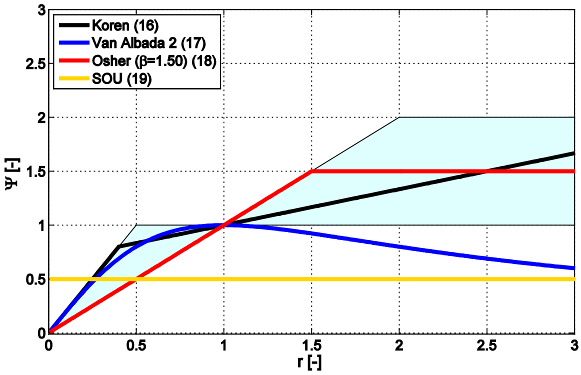


Figure 12: TVD region and limiter function for limiters with id 16-19.

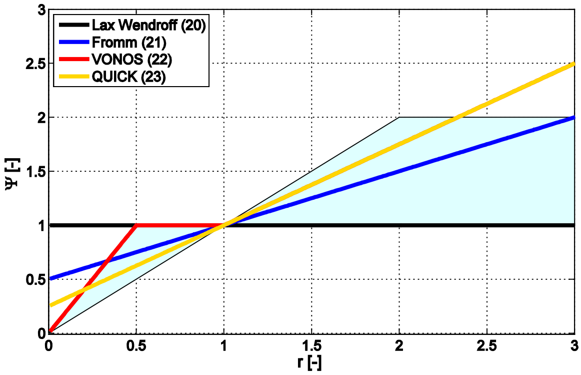


Figure 13: TVD region and limiter function for limiters with id 20-23.

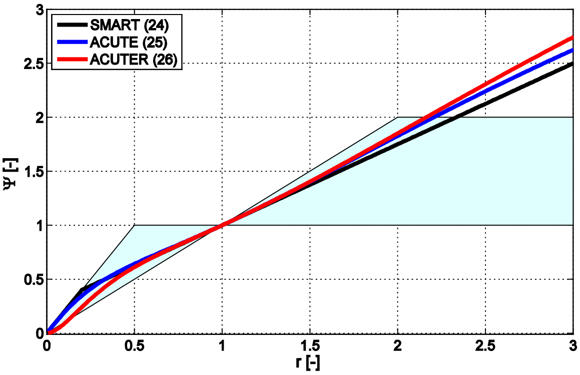


Figure 14: TVD region and limiter function for limiters with id 24-26.

The higher order flux limiters are only used when all three variables that are present in the calculation comes from bubble-bubble borders (upwind velocity) or bubble sections (upwind mass or specific enthalpy). If plugs are involved, a first order upwind value is used. The reason for this is that a slug might typically be much longer than the order control volumes involved in the calculation.

The water-faucet test-case as proposed by [31] was tested for the limiters presented in Table 1 to check their performance and to check that they had been implemented correctly. This test case consists of a 12 m long -90° inclined pipe with an inner diameter of 1 m. The initial conditions are $\alpha_l = 0.8$ and $u_l = 10$ m/s, with stagnant gas. The liquid velocity at the inlet is 10 m/s, while the gas flow rate is zero. The wall and interfacial friction is turned off, and the gas and liquid densities are 1.16 and 1000 kg/m3 respectively. The liquid is incompressible, and the gas compressibility was set to 1e-8 kg/m3/Pa to give an approximately incompressible gas phase. Because of the effect from gravity the initial column of liquid with a holdup of 0.8 will stretch and become thinner, with a volume fraction discontinuity propagating towards the exit.

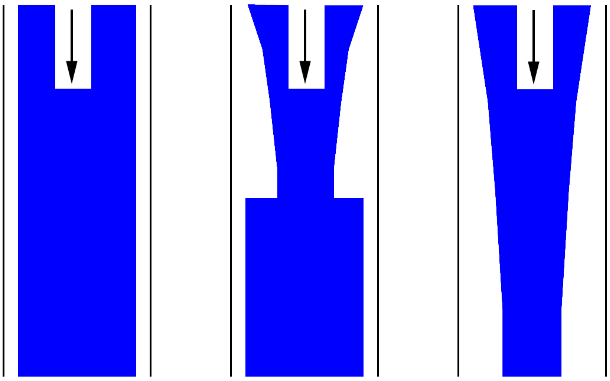


Figure 15: Illustration of the water-faucet test case.

This problem has the following analytical solutions for the liquid holdup and liquid velocity:

$$\alpha_l = \begin{cases} \dfrac{\alpha_{l0} u_{l0}}{\sqrt{2g\left(x-x_0\right)+u_{l0}^2}} & x \le x_0 + u_{l0}t + 0.5gt^2 \\ \alpha_{l0} & \text{otherwise} \end{cases} \tag{27}$$

$$u_l = \begin{cases} \sqrt{u_{l0}^2 + 2g\left(x-x_0\right)} & x \le x_0 + u_{l0}t + 0.5gt^2 \\ u_{l0} + gt & \text{otherwise} \end{cases} \tag{28}$$

Here x_0 is in the middle of the inlet section (at $-0.5\Delta x$), while u_{l0} and α_{l0} are 10 m/s and 0.8 respectively. The simulated and theoretical values for the gas void fraction at time 0.33 s are shown for simulations with 128 and 1024 grid points in Figure 16 and Figure 17. The limiters have been plotted sorted on the average deviation from the analytical solution, in decreasing order. The default semi-implicit discretization of the Sluggit code was used, with backward Euler time integration.

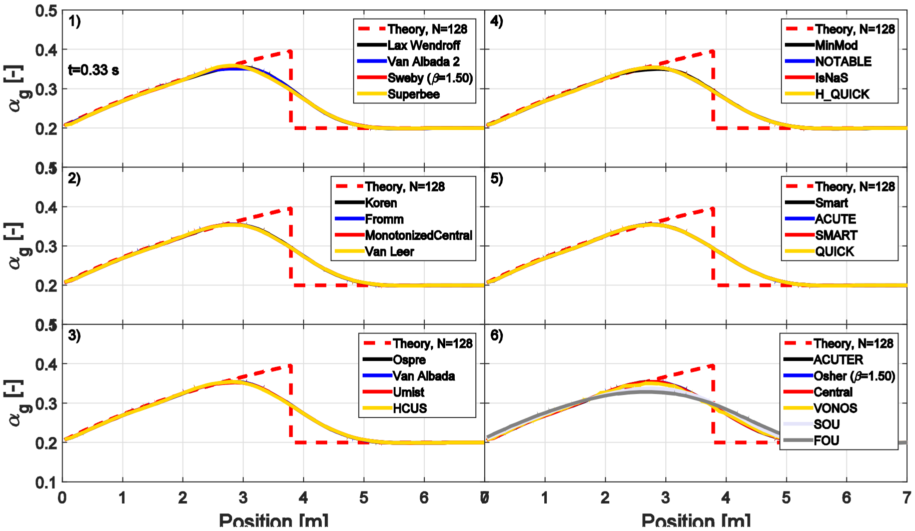


Figure 16: Gas void fraction from the simulations and from theory with the different flux limiters for the water-faucet test case for 128 grid points, at time 0.33 s.

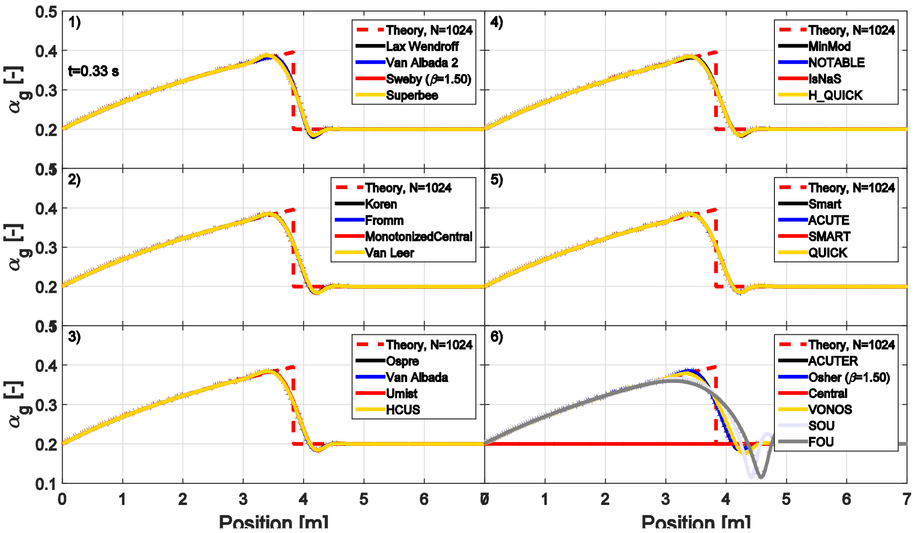


Figure 17: Gas void fraction from the simulations and from theory with the different flux limiters for the water-faucet test case for 1024 grid points, at time 0.33 s.

It can be difficult to distinguish the results for the different limiters in Figure 16 and Figure 17, except for the first and second order upwind schemes and the VONOS limiter in the bottom right plot which gives a significantly worse result than the rest. The accuracy of a limiter will be affected by the truncation error and is not simply given by the formal order of accuracy, and is most likely somewhat dependent on the selected test case. The results will also be dependent on the discretization of the framework in which they are implemented. It can however be seen that increasing the grid density from 128 to 1024 grid points, the simulation results are closer to the theoretical value. Further extensive testing would probably be required to distinguish the performance of the different limiters, perhaps also using more difficult test cases as robustness is just as important as the accuracy. The results at least indicates that the implementation of the different limiters seems to work well in the Sluggit framework, and that the higher order limiters gives better results than the basic first order upwind scheme as expected.

2.6.1 Special upwind coefficients

For certain cases the upwind coefficients are not calculated according to equation (16). For bubble-plug or plug-bubble borders, the upwind coefficients are set to zero and one respectively, to make the upwind specific mass point to that of the slug. This is done for both slug fronts and slug tails, and for pigs. This is because the velocities at the slug border belongs to the slug, and consequently the specific mass of the slug should be used to get a consistent mass flux. The same applies for a closed inlet or outlet border, where the upwind coefficients are set to one and zero respectively.

These hardcoded upwind coefficients are however only used for the specific mass, while upwind velocities and enthalpies needed for the convection of momentum and energy should be calculated normally according to equation (16) and (17).

Note that for slugs the use of hardcoded upwind coefficients should only be done for the continuous oil and water fields (*ol* and *wl*). For the entrained fields one should calculate the upwind coefficients as normal. The only purpose of the forced upwind approach is that the slug velocity is associated with the correct phase fractions at the slug borders.

2.7 Time integration schemes

The default numerical scheme in the Sluggit code for time integration is the implicit first order backward Euler method (BDF1), though the second order accurate Crank-Nicolson/trapezoidal (CN) and Backwards Euler method (BDF2) can also be used. These three methods are all unconditionally stable and can be presented by equation (29), showing the time integration $\dot{y} = F\left(y\right)$:

$$\frac{a_0 y^{n+1} + a_1 y^n + a_2 y^{n-1}}{\Delta t} = \left(1-\Theta\right) F\left(y^{n+1}\right) + \Theta F\left(y^n\right) \tag{29}$$

Table 2 shows the different coefficients for the different time integration schemes.

Table 2: Coefficients for the different time integration schemes.

Method	a_0	a_1	a_2	Θ
BDF1	1	-1	0	1
BDF2	1.5	-2	0.5	1
CN	1	-1	0	0.5

Note that both the BDF2 and CN methods needs a fully implicit discretization for the entire momentum equation to be second order accurate, and that the CN method is only unconditionally stable for $0.5 \le \Theta \le 1$. The default discretization in the Sluggit code is semi-implicit as this gives a relatively simple analytical Jacobian, but using a fully implicit discretization is also possible. The Jacobian is then calculated as for the semi-implicit discretization, and is updated with the newest available values in the iteration procedure. Ideally the correct Jacobian should have been calculated immediately, so convergence issues might thus happen in certain cases with the fully implicit schemes. It is however possible to calculate the Jacobian numerically using finite differences, though this is somewhat slower. Using the BDF2 method with the Sluggit code is nevertheless problematic since it requires values from two time levels back, and it is not straightforward what these values should be after having performed grid management operations. The Crank-Nicolson method should however be fine to use as it only uses values from the previous time step, but this has not been tested much when plugs and grid management is involved. It should however work fine when only bubble sections are involved, like in the test cases presented in Paper 4. An improved version of the BDF2 scheme also exists as shown by [32], and is obtained by taking one half of the BDF2 and BDF3 coefficients. In this thesis, we refer to this method as the BDF2.5 method. This method includes one more time level than BDF2, and yields the optimal A-stable and L-stable BDF method. The BDF2 and BDF2.5 methods have only been used in Paper 4, and the BDF2.5 method is only available in the Matlab model used for the simulations in Paper 4.

The CN method is however not L-stable, meaning that it does not damp unresolved high frequency waves. This means that it can give oscillations in the pressure, unless the time step is small enough to resolve the pressure waves (which is typically not the case). Testing with the simulations presented in Paper 4 showed that a value of 0.5 for Θ would typically give unstable results, and that a value 0.55 seemed much better to use. The CN method has a lower truncation error than BDF2 when Θ is 0.5, but in the test cases in Paper 4 BDF2 becomes slightly more accurate if using a value of 0.55. The local truncation errors for the CN and BDF2 scheme are shown in equation (30).

$$
\begin{aligned}
&\text{CN} && \left(\Theta-\frac{1}{2}\right)\Delta t^2 F'' + \left(\frac{\Theta}{2}-\frac{1}{3}\right)\Delta t^3 F''' + O\left(\Delta t^4\right) \\
&\text{BDF2} && -\frac{\left(r_{\Delta t}+1\right)^2}{6 r_{\Delta t}\left(2 r_{\Delta t}+1\right)}\Delta t^3 F''' + O\left(\Delta t^4\right)
\end{aligned}
\tag{30}
$$

Here, $r_{\Delta t}$ is the ratio of the previous and current time step. When the time step is constant, the leading coefficient of the truncation error of BDF2 becomes $-\frac{2}{9}$. It is interesting to note that an increase in the time step causes the leading coefficient of the truncation error of BDF2 to decrease, while a decrease in time step increases the coefficient. It is also interesting to see that the coefficient in the limit of $r \to \infty$ goes to $-\frac{1}{12}$, which is the same coefficient as that of Crank-Nicolson with $\Theta = 0.5$. This is also the optimal possible value for an implicit A-stable method (second Dahlquist barrier).

Note that the BDF2 method suffers from a start-up problem, as it uses values from two time steps back. Another method must thus be used in the first time step. Furthermore, the coefficients for BDF2 as presented in Table 2 are only valid for a constant time step. When the time step is not constant, the coefficients becomes functions of the previous time step values, see for instance [33-36] who all obtain the same solution:

$$\begin{aligned} a_0 &= \frac{2r_{\Delta t}+1}{r_{\Delta t}+1} \\ a_1 &= -\left(r_{\Delta t}+1\right) \\ a_2 &= \frac{r_{\Delta t}^2}{r_{\Delta t}+1} \\ r_{\Delta t} &= \frac{\Delta t^{n+1}}{\Delta t^n} \end{aligned} \tag{31}$$

Furthermore, as shown by [36], the ratio $r_{\Delta t}$ must be smaller than $1+\sqrt{2}$ for the BDF2 method to be convergent and stable (for ODE's), while [37] reports an upper bound for PDE's of approximately 1.91. Testing with the simulations presented in Paper 4 however showed no problems using higher ratios than this. The coefficients for the BDF2.5 scheme are also functions of the time step ratios. This derived in the section presenting Paper 4.

The stability of the different time integration schemes can be illustrated by investigating the test equation

$$\dot{y} = \lambda y \tag{32}$$

where λ can be complex. The exact solution to this equation ($y = e^{\lambda\Delta t}$) will grow in time if $\operatorname{Re}(\lambda\Delta t) > 0$, and decay if $\operatorname{Re}(\lambda\Delta t) < 0$. A method that has the property that $|G| < 1$ for $\operatorname{Re}(\lambda\Delta t) < 0$ is called an *A-stable* method.

Furthermore, we can define that amplification *G* from one time step to the next as:

$$G = \left| \frac{y^{n+1}}{y^n} \right| \tag{33}$$

The solution for *G* for the test equation is a second order polynomial:

$$G^2 \left(a_0 - \lambda \Delta t \right) + G \left(a_1 - \lambda \Delta t \left(1 - \Theta \right) \right) + a_2 = 0 \tag{34}$$

Methods that in addition has the property that $G \to 0$ as $\mathrm{Re}\left(\lambda \Delta t\right) \to -\infty$ are called *L-stable*. This means, that unresolved high frequency oscillations will be damped. Both the BDF1 and BDF2 methods are L-stable, but Crank-Nicolson is not ($G \to 1$ as $\mathrm{Re}\left(\lambda \Delta t\right) \to -\infty$). This means that the Crank-Nicolson method is prone to oscillations in the pressure since the time step associated with resolving the acoustic pressure waves is much smaller than the typical time steps used with the two-fluid model. Contour plots of the amplification *G* for the different time integration methods are shown in Figure 18. If *G* is less than 1, the solution will be damped in time.

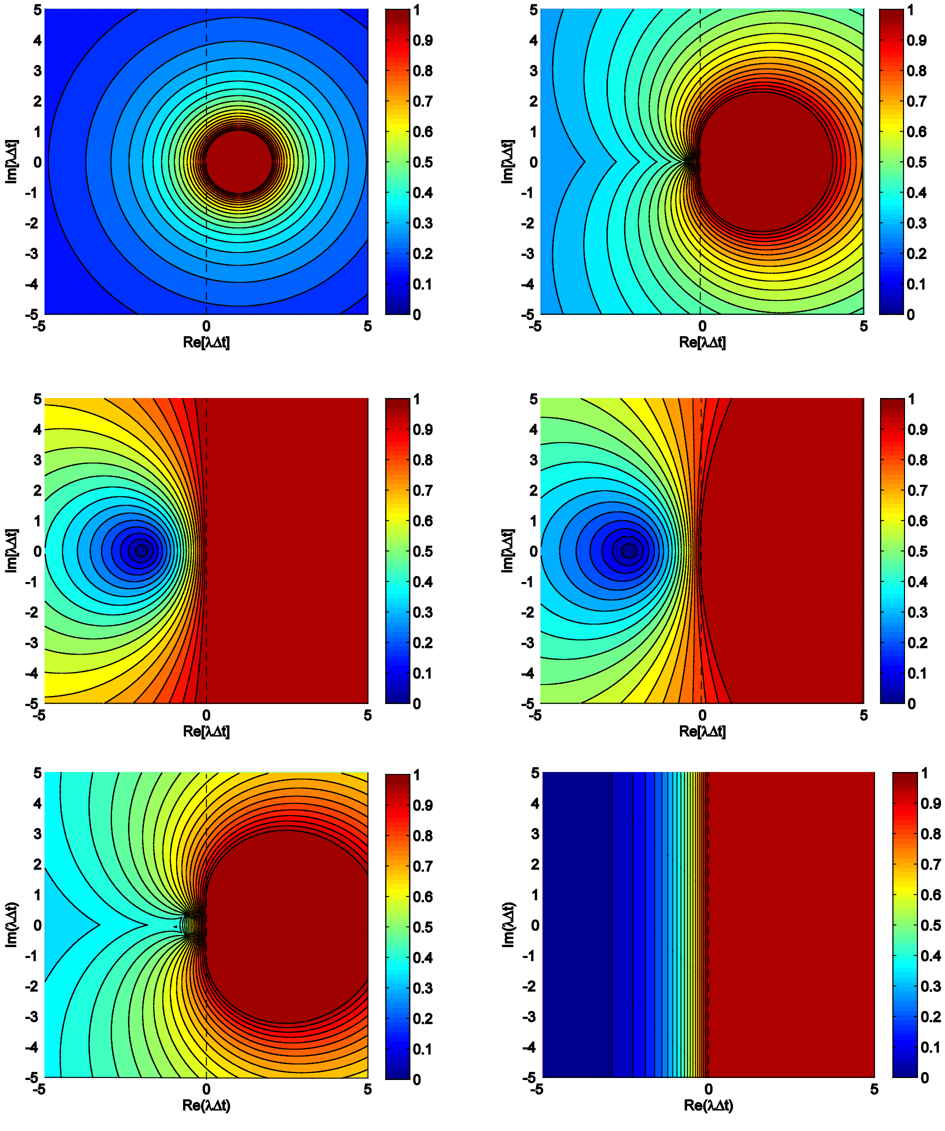


Figure 18: Stability plots of the BDF1, BDF2, CN and BDF2.5 time integration methods. The plots at the top shows from left to right the BDF1 and BDF2 methods, while the middle plots shows the CN method with $\Theta = 0.5$ and $\Theta = 0.55$ respectively. The lower left plot shows the BDF2.5 method, and the lower right plot shows the exact solution. |G|>1 is shown as a single colour.

Depending on the axis limits, the plots shown in Figure 18 can however give a false perception that the CN method gives a much more correct amplification compared to the BDF2 method. The larger the axis limits, the more pronounced this difference will seem. For slug capturing however, the time step is typically controlled by the CFL criterion for the gas and liquid velocities, which will result in $\lambda\Delta t = \frac{\lambda\Delta x}{u_{\max}} < \Delta x$. This also means that the finer the grid, the closer to the origin the value of $\lambda\Delta t$ will lie. Zooming in closer around the origin gives a different perspective of the BDF2 method, as shown in Figure 19. Note also that a red colour in the right half plane just means an amplification larger than 1 (since $G > 1$ is limited to 1), and does not necessarily mean the same amplification as the exact solution. This is illustrated in Figure 19, where the colour shows the deviation from the exact solution instead of the magnitude of *G*.

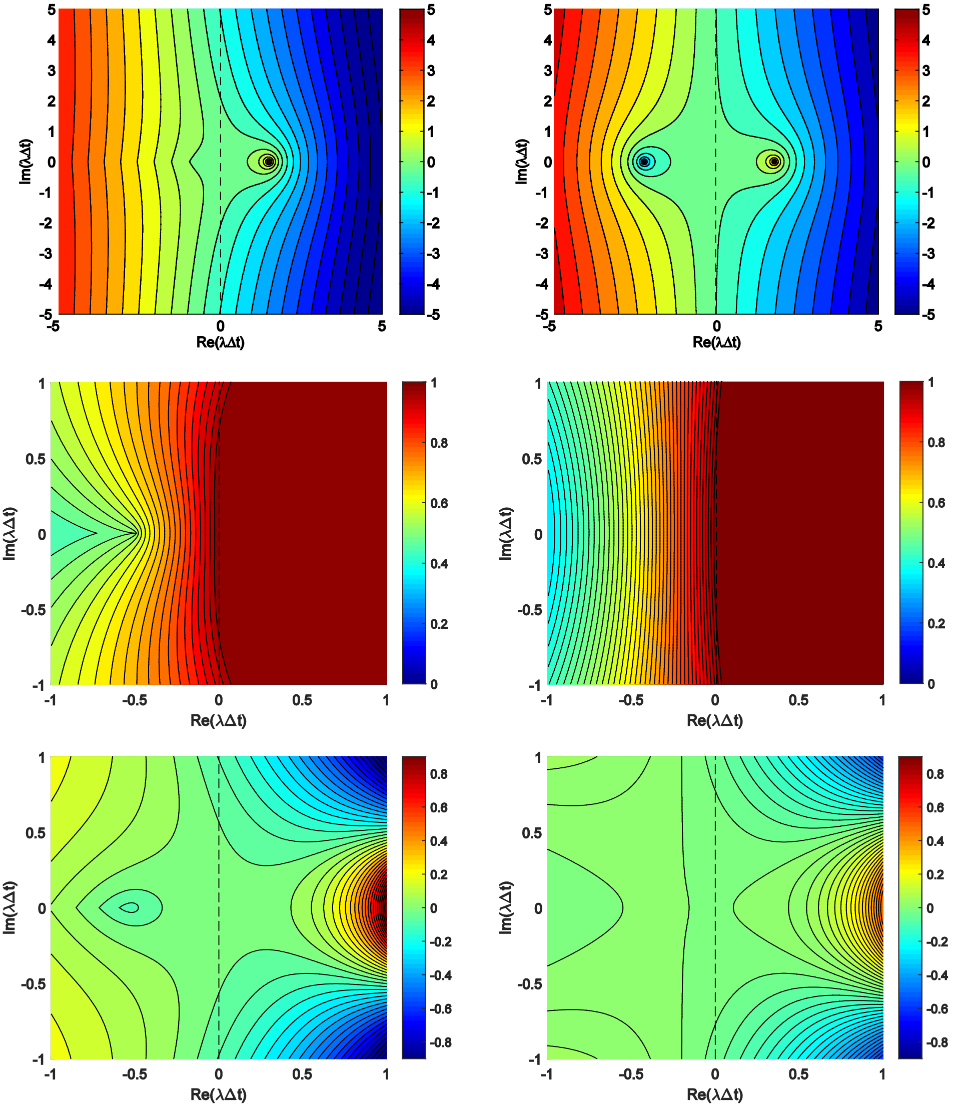


Figure 19: The amplification factor for BDF2 (left column) and the Crank-Nicolson method with θ = 0.55 (right column). In the first row the colour shows log(G) − log(G_{exact})), while the second row shows G with reduced axis limits and with $G \geq 1$ given a single colour. The colour in the last row shows $G - G_{\text{exact}}$ with reduced axis limits.

Increasing Θ for the CN method can improve the L-stability, at the cost of somewhat reduced order of accuracy (see equation (30)). The amplification in the limit of L-stability for the CN method as function of Θ becomes $G = \frac{1-\Theta}{\Theta}$, and is plotted in Figure 20.

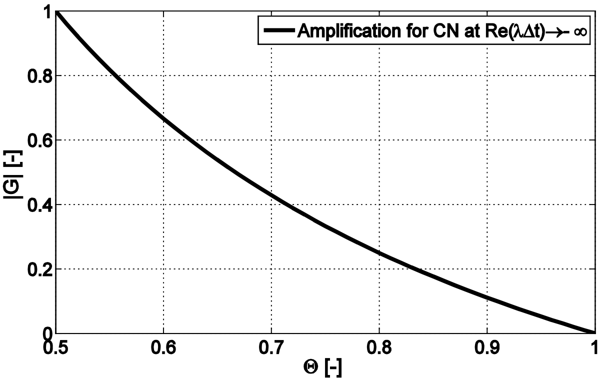


Figure 20: Amplification for the Crank-Nicolson method in the limit for L-stability as function of Θ, for the presented test equation.

As mentioned, the BDF2 scheme is dependent on the ratio of the previous and current time step. This does not only affect the leading coefficient in the truncation error, but also affects the stability region as shown in Figure 21. With an increase in the time step ($r > 1$), the region of correct instability in the right half planes is improved, and the truncation error decreases. However, the method is then no longer A-stable. Instead the method becomes so-called A(α)-stable, meaning that the instability region stretches somewhat into the left half plane (which is incorrect). The parameter α describes the angle at the stability boundary at the origin, with $\alpha = 90$ meaning a fully A-stable method. If the time step is decreased ($r < 1$), the region of correct instability in the right half plane decreases towards that of backward Euler, and the truncation error increases.

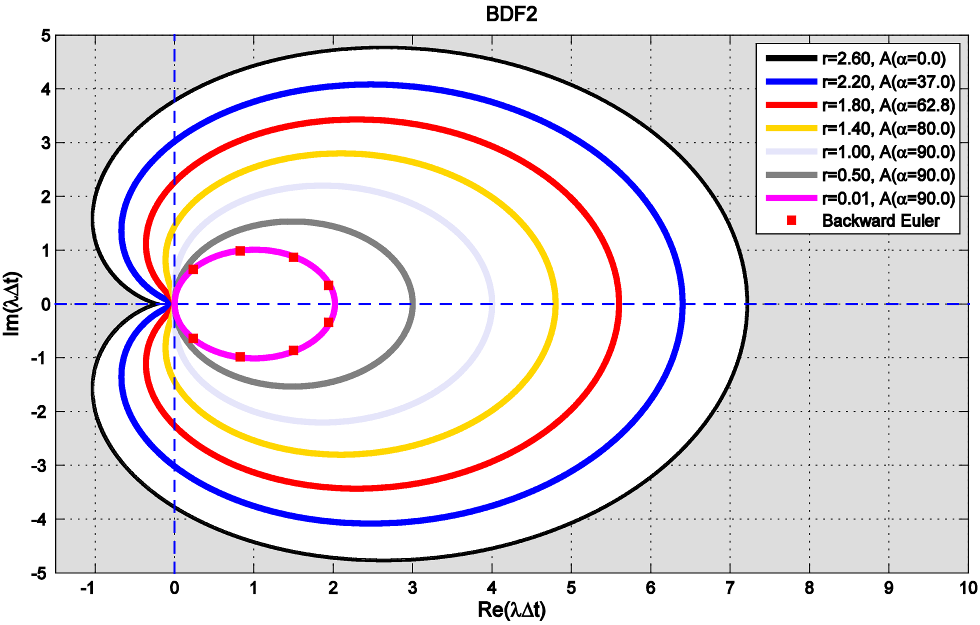


Figure 21: Stability region of BDF2 as function of the time step ratio *r*. The white area illustrates amplification larger than 1.

2.8 Pipe geometry

2.8.1 Walls

Different walls with multiple wall layers might be added to the pipeline, to simulate radial heat transfer. Each pipeline wall will be divided into a certain number of wall sections, calculated from a user-specified length (L_{wall}). If not specified, the grid coefficient that controls the minimum bubble section length is used instead to calculate the number of sections.

The number of wall sections n_{wallsecs} are then calculated by:

$n_{\text{wallsecs}} = \max\left(\text{round}\left(\frac{L_{pipe}}{L_{wall}}\right),1\right)$, where L_{pipe} is the length of the pipe.

Each wall section will have as many wall layers as there are walls defined for this pipe. It is possible to specify different wall options for different pipes in a simulation. One pipe might be without walls, another pipe might have one wall, and other pipes can have multiple wall layers. See section 2.9.5 for a description of the calculation of the wall heat transfer.

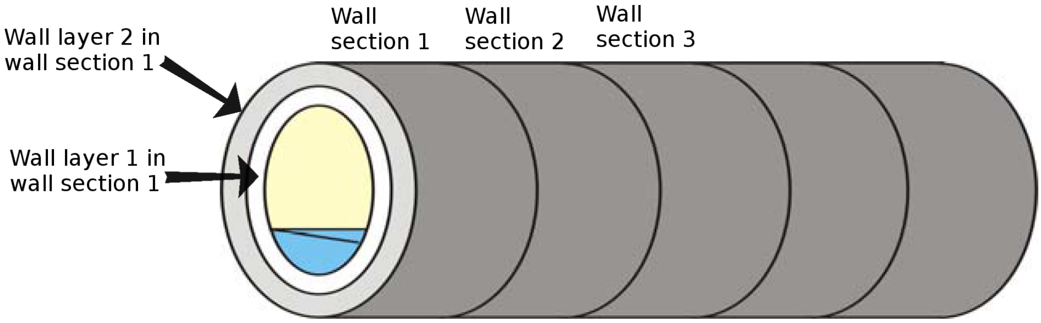

Wall layer 2 in wall section 1
Wall layer 1 in wall section 1
Wall section 1
Wall section 2
Wall section 3

Figure 22: Illustration of a single section in the computational grid, with one wall divided into 5 wall sections. Each wall section consists of two wall layers.

2.8.2 Curved pipe geometry

It is possible to use bent pipe sections in the Sluggit framework, as illustrated in Figure 23. To use a bent pipe in a simulation one must specify both a bend angle (in radians), and a bend radius (R_b). The length of the pipe segment will then be calculated as bend radius * bend angle. Just as for regular pipe segments, the angle at the start of the pipe must also be specified. This is the angle specified as θ_{in} in Figure 23.

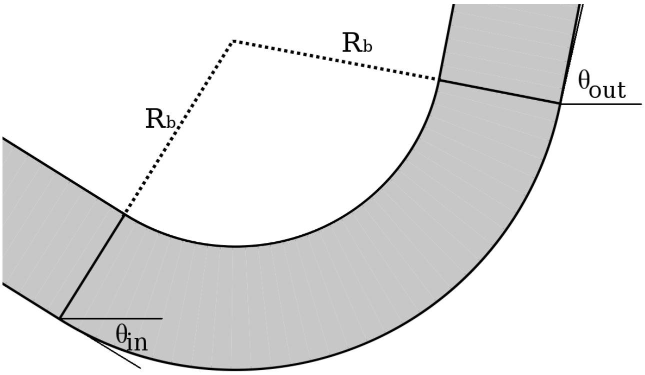


Figure 23: Bent pipe geometry.

For a bent pipe, the local inclination in the pipe is a function of position:

$$\theta(x) = \theta_{in} + \frac{s}{R_b}(x - x_0) \tag{35}$$

Here θ_{in} is the inclination (in radians) at the start of the bent pipe (at $x \equiv x_0$). The coefficient *s* equals 1 if the pipe has a low-point (is concave), and -1 if the pipe is convex. R_b is the bend radius of the pipe, and x_0 is the absolute location at the start of the pipe. For bent pipes one must then take into account that the inclination is a function of position when integrating the basic momentum equation over the momentum control volume. This is described in section 2.9.4.1. Testing however showed quite similar results when using bent pipe segments compared to just using straight pipe segments, so straight pipe segments has mainly been used in the simulations presented in this study. The bent geometry framework is however described and taken into account in the derivation of the momentum and energy equations, which involves integrals over the pipe inclination.

2.8.3 Geometrical relations

The wetted angle β_l of the stratified liquid layer is required to calculate among others the interfacial and wall perimeters, and can be found by solving the following equation:

$$\frac{\beta_l - \sin(\beta_l)\cos(\beta_l)}{\pi} - \alpha = 0 \tag{36}$$

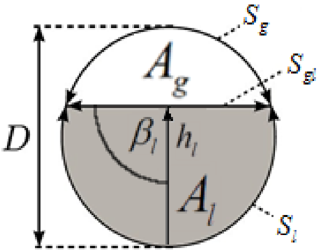


Figure 24: Geometrical relations in stratified flow.

Equation (36) is solved by Newton's method, using the Biberg approximation [38] shown in equation (37) as the initial guess.

$$\beta_{l,biberg} = \pi\alpha_l + \left(\frac{3\pi}{2}\right)^{\frac{1}{3}}\left(1 - 2\alpha_l + \alpha_l^{\frac{1}{3}} - \alpha_g^{\frac{1}{3}}\right) \tag{37}$$

The interfacial (*gl*), liquid (*l*) and gas (*g*) perimeters and liquid height for stratified flow are given by:

$$\begin{gathered} S_{gl} = D\sin\beta_l \quad S_l = D\beta_l \quad S_g = D\pi - S_l \\ h_l = \frac{D}{2}(1 - \cos\beta_l) \end{gathered} \tag{38}$$

For annular flow, the corresponding relations are:

$$S_{gl} = \begin{cases} \pi D\sqrt{1-\alpha_l} & \alpha_l == 0 \\ 0 & \alpha_l > 0 \end{cases}$$
$$S_l = \begin{cases} \pi D & \alpha_l > 0 \\ 0 & \alpha_l == 0 \end{cases} \tag{39}$$
$$S_g = \begin{cases} \pi D & \alpha_l == 0 \\ 0 & \alpha_l > 0 \end{cases}$$

The film thickness in annular flow is given by:

$$\delta_l = \frac{D}{2}\left(1-\sqrt{1-\alpha_l}\right) \tag{40}$$

2.9 Equations

2.9.1 Some general comments

2.9.1.1 Control volume integration

In the derivation of the finite volume representation of the mass, momentum, pressure and energy equations, the basic conservation equations are integrated over the control volume. In the Sluggit framework, a control volume might span different pipes, with different properties like cross sectional area or inclination. To take into account the possibility that the different pipes might have different properties, the integration over a control volume is split up into a sum of integrals (split at the *static borders*, which are the borders that separates different pipe segments. This is illustrated for a slug control volume in Figure 25. For a general property Φ, this can be written as:

$$\int_{V_k} \Phi dV = \int_{A_k} \int_a^f \Phi dA dx = A_k \left(\int_a^b \Phi dx + \int_b^c \Phi dx + \int_c^d \Phi dx + \int_d^e \Phi dx + \int_e^f \Phi dx \right) \qquad (41)$$

We refer to the different parts within the control volume as *sub control volumes.* For instance, the slug control volume shown in Figure 25 consists of five sub control volumes.

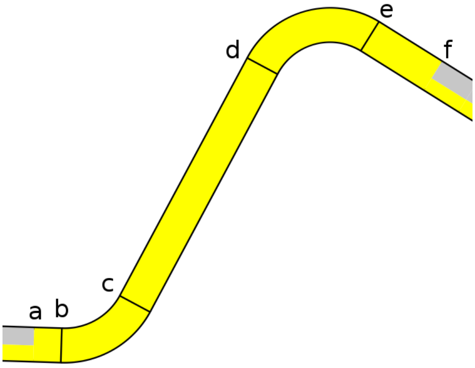


Figure 25: The integration locations for a slug control volume. The integral over the slug control volume can be split into integrals from positions a-b, b-c, c-d, d-e and e-f.

$A_{k,j}$

2.9.1.2 Cross sectional area

The cross sectional pipe area is needed many places in the equations to be solved, amongst others in the mass and momentum fluxes. For borders located at the junction between two different pipes, the *minimum* area of the area from the pipe to the left and to the right of the border is used. For most borders, the pipe to the left and right will be the same pipe. For static borders however this will not be the case, as these borders separate different pipes. Unless specified otherwise, any cross sectional border area is the minimum of the left and right areas. Note that the diameter and perimeter at static borders should also then be the minimum value.

2.9.1.3 Average velocity in a control volume

Since the momentum control volume might consist of pipes with different cross sectional areas, the velocity we solve for will be the average velocity within the control volume. This can also be defined as the volume flow in the control volume divided by the average cross sectional area:

$$u_{k,j} = \frac{Q_{k,j}}{\overline{A}_{k,j}} \tag{42}$$

It is here assumed that the density is constant within the control volume, which gives a constant volume flow. The average cross sectional area is defined by:

$$\overline{A}_{k,j} = \frac{V_{k,j}}{L_j} = \frac{V_j \alpha_k}{L_j} = \overline{A}_j \alpha_k \tag{43}$$

We can use the following relation between the volume flow at a local point within the control volume, and the volume flow through the average area ($\overline{A}_{k,j}$) in the control volume to find the velocity at any point:

$$\begin{aligned} u_{k,j,local} A_{k,j,local} &= u_{k,j} \overline{A}_{k,j} \\ u_{k,j,local} &= u_{k,j} \frac{\overline{A}_{k,j}}{A_{k,j,local}} \end{aligned} \tag{44}$$

Since the phase fraction is the same everywhere inside a section even if it spans multiple pipes, we can simplify further and write:

$$u_{k,j,local} = u_{k,j}\frac{\overline{A}_{k,j}}{A_{k,j,local}} = u_{k,j}\frac{\overline{A}_{j}\alpha_{k,j}}{A_{j,local}\alpha_{k,j}} = u_{k,j}\frac{\overline{A}_{j}}{A_{j,local}} \tag{45}$$

To simplify, we call this area ratio for a_j^*:

$$a_j^* = \frac{\overline{A}_j}{A_{j,local}} \tag{46}$$

For an example, see Figure 7 where the local area at the border represented by index j+1 is different than the control volume area for the same border. Assuming no border movement, the mass flux though this border is defined as $m_{k,j+1}A_{j+1}u_{k,j+1,local}$. This mass flux is present in the pressure equation, but we do not solve for $u_{k,j+1}$ and convert to the control volume velocity instead which is the velocity we solve for: $m_{k,j+1}A_{j+1}u_{k,j+1}\frac{\overline{A}_j}{A_{j,local}}$

For incompressible plugs, equation (45) will be exact and has to be used since the density is constant and the volume flow has to be the same everywhere. This scaling will be used wherever the local velocity is needed (like in the momentum fluxes), to convert to the velocity through the control volume which is the variable we solve for.

2.9.2 Mass equation

The mass/continuity equation integrated over the control volume for sub-field *f* reads:

$$\frac{dM_f}{dt}+\oint_{A_{bf}}\rho_f\left(\boldsymbol{u}_f-\boldsymbol{u}_b\right)\cdot\boldsymbol{n}_f dA_{bf}=\dot{M}_f^{src} \tag{47}$$

Some illustrations of the different control volumes can be found in section 2.3.
The term $\dot{M}_{f,j}^{src}$ represents the contributions from possible mass sources or sinks, and the total effect of all types of entrainment and deposition.
The discretized mass equation reads:

$$\frac{M_{f,j}^{n+1}-M_{f,j}^{n}}{\Delta t}=\hat{m}_{f,J-1}^{n+1}A_{J-1}^{n}\left(u_{f,J-1}^{n+1}-u_{b,J-1}^{n+1}\right)-\hat{m}_{f,J}^{n+1}A_J^n\left(u_{f,J}^{n+1}-u_{b,J}^{n+1}\right)+\dot{M}_{f,j}^{src} \tag{48}$$

The BDF2 and Crank-Nicolson time integration methods can also be used for the mass equation, see equation (29).

In equation (48), *V* the total cell volume, $V_f=\alpha_f V$ is the fluid volume, M_f the fluid mass, *A* the pipe cross sectional area, u_f the fluid velocity, u_b the border velocity, while *m* is the specific mass defined as:

$$m_f=\rho_f\alpha_f=\frac{\rho_f V_f}{V}=\frac{M_f}{V} \tag{49}$$

The terms ρ_f and α_f represents the fluid density and phase fraction respectively.
The mass equation is solved for all sections, regardless of type.

2.9.3 Pressure equation

The mass and momentum equations for gas and liquid results in as many equations are there are unknown masses and velocities. The new pressure in the momentum equation is however also unknown, and consequently another equation is needed to close the system of equations. The pressure equation is obtained by combining the mass equations, an equation of state of the form $\rho = \rho\left(p,T\right)$, and using the fact that the phase fractions should sum to 1. The transient term in the mass equation (equation (47)) is first expanded by the product rule:

$$\frac{dM_f}{dt} = \frac{d\left(\rho V\right)_f}{dt} = \left[\left(\frac{\partial \rho_f}{\partial p}\right)_{T_f}\frac{\partial p_f}{\partial t} + \left(\frac{d\rho_f}{dT}\right)_p \frac{dT_f}{dt}\right]V_f + \rho_f \frac{dV_f}{dt} = \dot{M}_f^{src} - \oint_{A_{bf}} \rho_f \left(\boldsymbol{u}_f - \boldsymbol{u}_b\right)\cdot \boldsymbol{n}_f dA_{bf} \tag{50}$$

Dividing the resulting equation by the density ρ_f and summing over all sub-fields, gives the pressure equation:

$$\sum_k \left(\frac{V_k}{\rho_k}\left[\left(\frac{\partial \rho_f}{\partial p}\right)_{T_f}\frac{dp_f}{dt} + \left(\frac{\partial \rho_f}{\partial T}\right)_p \frac{dT_f}{dt}\right] + \frac{dV_f}{dt} + + \frac{1}{\rho_f}\oint_{A_{bf}} \rho_f \left(\boldsymbol{u}_f - \boldsymbol{u}_b\right)\cdot \boldsymbol{n}_f dA_{bf} \right) = \sum_f \frac{\dot{M}_f^{src}}{\rho_f} \tag{51}$$

By carrying out the summation of the volume derivative $\sum_f \frac{dV_f}{dt} = \frac{dV}{dt}$, we get:

$$\sum_f \left(\frac{V_f}{\rho_f}\left[\left(\frac{\partial \rho_f}{\partial p}\right)_{T_f}\frac{dp_f}{dt} + \left(\frac{\partial \rho_f}{\partial T}\right)_p \frac{dT_f}{dt}\right]\right]\right) + \frac{dV}{dt} + \sum_f \frac{1}{\rho_f}\oint_{A_{bf}} \rho_f \left(\boldsymbol{u}_f - \boldsymbol{u}_b\right)\cdot \boldsymbol{n}_f dA_{bf} = \sum_f \frac{\dot{M}_f^{src}}{\rho_f} \tag{52}$$

Using the Reynold's transport theorem for a variable Υ:

$$\frac{d}{dt}\int_{V^*} \Upsilon dV = \int_{V^*} \frac{\partial \Upsilon}{\partial t} dV + \int_{A^*} \Upsilon \mathbf{n} \cdot \mathbf{u}_b dA \tag{53}$$

The time derivative of the volume in equation (52) can be written as:

$$\frac{dV}{dt}=\frac{d}{dt}\int_{V^*}1dV=\overbrace{\int_{V^*}\frac{\partial(1)}{\partial t}dV}^{0}+\int_{A^*}\mathbf{n}\cdot\mathbf{u}_b dA=\left(A_J^n u_{b,J}^{n+1}-A_{J-1}^n u_{b,J-1}^{n+1}\right) \tag{54}$$

We then get:

$$\begin{aligned}&\frac{p_j^{n+1}-p_j^n}{\Delta t}\sum_k\frac{V_{f,j}^{n+1}}{\rho_{f,j}^n}\left(\frac{\partial\rho_{f,j}}{\partial p}\right)_{T_f}+\sum_k\frac{V_{f,j}^{n+1}}{\rho_{f,j}^n}\left(\frac{\partial\rho_{f,j}}{\partial T}\right)_p\frac{dT_f}{dt}+\left(A_J^n u_{b,J}^{n+1}-A_{J-1}^n u_{b,J-1}^{n+1}\right)\\&+\sum_f\frac{1}{\rho_{f,j}^n}\left[\hat{m}_{f,J}^{n+1}A_J^n\left(u_{f,J}^{n+1}-u_{b,J}^{n+1}\right)-\hat{m}_{f,J-1}^{n+1}A_{J-1}^n\left(u_{f,J-1}^{n+1}-u_{b,J-1}^{n+1}\right)\right]=\sum_f\frac{\dot{M}_{f,j}^{src}}{\rho_{f,j}^n}+\Psi_{f,j}^{n+1}\end{aligned} \tag{55}$$

A source term Ψ_j^{n+1} has also been included to correct for the volume fraction error from the last time step (see section 2.5).

The pressure equation can be rewritten to yield an expression for the new pressure, as function of amongst others the old pressure and the new velocities.

We then get:

$$\begin{aligned}&p_j^{n+1}-p_j^n=\Delta p_j=\\&\Delta t\frac{\left(\begin{aligned}&\sum_f\frac{1}{\rho_{f,j}^n}\left[\hat{m}_{f,J-1}^{n+1}A_{J-1}^n\left(u_{f,J-1}^{n+1}-u_{b,J-1}^{n+1}\right)-\hat{m}_{f,J}^{n+1}A_J^n\left(u_{f,J}^{n+1}-u_{b,J}^{n+1}\right)+\dot{M}_{f,j}^{src}\right]+\\&\Psi_{f,j}^{n+1}-\left(A_J^n u_{b,J}^{n+1}-A_{J-1}^n u_{b,J-1}^{n+1}\right)-\sum_f\frac{V_{f,j}^{n+1}}{\rho_{f,j}^n}\left(\frac{\partial\rho_{f,j}}{\partial T}\right)_p\frac{\Delta T_f}{\Delta t}\end{aligned}\right)}{\sum_f\frac{V_{f,j}^{n+1}}{\rho_{f,j}^n}\left[\left(\frac{\partial\rho_{f,j}}{\partial p}\right)_T\right]}\end{aligned} \tag{56}$$

The BDF2 and Crank-Nicolson time integration methods can also be used for the pressure equation. Note that the mass equation written in the form of equation (29) should then be the starting point for the derivation of the pressure equation.

For BDF2 this will result in the following additional terms:

$$\frac{\Delta t}{\sum_f \frac{V_{f,j}^{n+1}}{\rho_{f,j}^n}\left[\left(\frac{\partial \rho_{f,j}}{\partial p}\right)_T\right]}\sum_f \frac{1}{\rho_f^n}\left(-\left(a_1+a_0\right)M_f^n - a_2 M_f^{n-1}\right) \tag{57}$$

The coefficients a_0, a_1 and a_2 are defined in equation (29) .

This was found to give identical results to simulations where no pressure equation was used, using instead that the phase fractions should sum to one as a constraint to close the system of equations together with a numerical Jacobian. In other words, the extra term presented in equation (57) seems to be correct. If the Crank-Nicolson method is to be used, any terms originating from the time derivative (volume time derivative and the term including the temperature) should not be multiplied by the implicit fraction Θ (see equation (29)).

Equation (56) now contains the new sub-field velocities (u_f^{n+1}), while we want to solve for the new mixture velocities (u_k^{n+1}). We thus express the new sub-field velocities by a linear slip-relation with the new mixture velocities:

$$u_f^{n+1} = C_{f,slip}^n u_k^{n+1} + U_{f,slip}^n \tag{58}$$

See section 2.10.3 for a description of the coefficients in the slip relation.
The new border velocity is also related to the mixture velocities by a linear relationship:

$$u_{b,j}^{n+1} = C_{0l,j}^n u_{l,j}^{n+1} + C_{0g,j}^n u_{g,j}^{n+1} + U_{0,j}^n \tag{59}$$

See section 2.12 for a description of the border velocity coefficients.
The new volume and the mass flux terms containing the new masses are evaluated using the new values for section volume and mass, which are kept and updated in each iteration on the system of equations. These terms will thus be converging to the correct final value in each iteration. The iteration procedure is described in more detail in section 2.14.

It should be noted that there is an assumption in the pressure equation that the new density can be calculated as:

$$\rho_f^{n+1} = \rho_f^n + \frac{\partial \rho_f}{\partial p}\Delta p + \frac{\partial \rho_f}{\partial T}\Delta T_f \tag{60}$$

One can in other words get a slightly different density by using equation (60) than what one would get from using the equation of state directly. Consequently, the fluid densities are updated using equation (60) in the iteration procedure, to ensure convergence of the volume fraction error. However, after the iteration procedure is finished the densities are calculated directly from the equation of state. Depending on the equation of state, this might introduce a small amount of volume fraction error. For a linear equation of state however the results should be exactly the same. This issue is illustrated with an artificial equation of state (could be the pressure vs density points from a PVT-file), as shown in Figure 26. The temperature is assumed constant in this example.

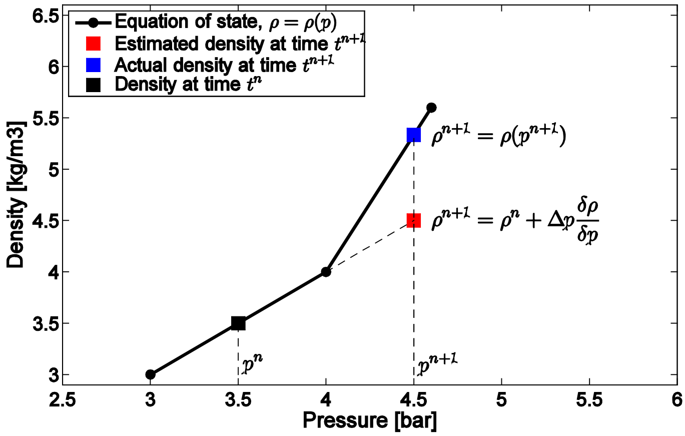


Figure 26: Illustration of an artificial equation of state, showing density calculated using the pressure derivative and the density calculated directly from the equation of state.

2.9.3.1 Area changes

If a border moves past different cross sectional areas during a time step, the pressure equation as presented in equation (56) will not be correct, due to several reasons that will be discussed in the following sections.

2.9.3.1.1 Volume time derivative

The terms that originate from the volume derivative will not be correct when a border moves through a region with area changes. Equation (61) shows how this should be calculated for *one* of the borders in a section. To account for changes in the cross-sectional area, we can use that the average cross-sectional area multiplied by the average border velocity during the time step should equal the volume the border will move past:

$$\frac{dV}{dt} = \overline{u}_b^{\,n+\frac{1}{2}}\,\overline{A}^{\,n+\frac{1}{2}} \tag{61}$$

Here we have defined the average area the border moves by as:

$$\overline{A}^{\,n+\frac{1}{2}} = \frac{V_b^{\,n+\frac{1}{2}}}{\Delta x_b^{\,n+\frac{1}{2}}} \tag{62}$$

$\Delta x_b^{\,n+\frac{1}{2}}$ is the distance covered by this border during this time step, $\overline{u}_b^{\,n+\frac{1}{2}}$ is the average border velocity during this time step, while $V_b^{\,n+\frac{1}{2}}$ is the total volume the border has moved by from time t^n to t^{n+1}. The time index $n+\frac{1}{2}$ is here used to indicate a property which is defined between time *n* and *n*+1.

The problem now is that we do not know the distance and volume that the border will move by until we have solved the momentum equations. This can however be incorporated as part of the iteration procedure described in section 0, by storing the average area a border moves at the end of each volume fraction iteration.

The average border velocity during the time step is also unknown until we have solved the momentum equations, and must also be a part of the iteration procedure. To be able to relate the border velocity in the volume time derivative term implicitly to the new mixture velocity, we define the average border velocity during one time step as the new border velocity times a scaling constant:

$$\overline{u}_b^{\,n+\frac{1}{2}} = u_b^{n+1}\omega_b \tag{63}$$

The scaling constant is updated at the end of each volume fraction iteration, and is defined as:

$$\omega_b = \frac{\Delta x_b^{n+\frac{1}{2}}}{\Delta t}\frac{1}{u_b^{n+1}} \tag{64}$$

The volume time derivative term will then still be related implicitly to the new mixture velocities, and will converge towards the actual volume change per time:

$$\frac{dV}{dt} = \overline{u}_b^{\,n+\frac{1}{2}}\overline{A}^{\,n+\frac{1}{2}} = u_b^{n+1}\omega_b\frac{V_b^{n+\frac{1}{2}}}{\Delta x_b^{n+\frac{1}{2}}} = u_b^{n+1}\frac{\Delta x_b^{n+\frac{1}{2}}}{\Delta t}\frac{1}{u_b^{n+1}}\frac{V_b^{n+\frac{1}{2}}}{\Delta x_b^{n+\frac{1}{2}}} = \frac{V_b^{n+\frac{1}{2}}}{\Delta t}$$

2.9.3.1.2 Local velocity vs. control volume velocity

The pressure equation needs the mass flux through the end-points of the pressure control volume, which is the area at the borders. The new velocity through the area at the location of the borders are needed, but the velocity in the momentum control volume is defined through the average momentum control volume area and not necessarily at area where we need the mass flux. This is particularly important for incompressible plugs. In other words, the velocity in the mass flux that enters the equation is not necessarily the velocity solve for. To fix this we approximate the local velocity through the area where the mass flux is needed by an area-scaling constant, as described in section 2.9.1.3.

All mixture field velocities in the pressure equation will thus be multiplied by the area ratio factor a^* (see equation (46)), to convert the local velocities to control volume velocities.

To simplify the final expression of the pressure equation that takes into account area changes we also define a border velocity which includes the area-scaling factor:

$$u_{b,j}^{*n+1} = C_{0l,j}^{n} a_j^* u_{l,j}^{n+1} + C_{0g,j}^{n} a_j^* u_{g,j}^{n+1} + U_{0,j}^{n} \tag{65}$$

2.9.3.1.3 Slug border mass fluxes

The mass flux terms as they are formulated in equation (56) are not correct when a slug border moves past area changes (see section 2.12.6). As described in section 2.12, the slug border mass fluxes will be calculated differently when there are area changes, and this mass flux will be stored at the end of each time step. In the pressure equation however we want to relate the mass fluxes implicitly to the new velocities, so we keep the mass flux terms as shown in equation (56), but multiply them by a scaling constant (κ_f) which relates them to the actually used and stored slug border mass flux. This scaling constant is updated at the end of each volume fraction iteration, and is defined as:

$$\kappa_{f,j}^{n+1} = \frac{Q_{slug,k,j}^{n+1}}{A_j^n \left(C_{slip,f,j}^{n} a_j^* u_{k,j}^{n+1} + U_{slip,f,j}^{n} - u_{b,j}^{*n+1} \right)} \tag{66}$$

Here $Q_{slug,k,j}^{n+1}$ is the actually used volume flux for the slug, as described in section 2.12.

2.9.3.1.4 Final form of the pressure equation

The final form of the pressure equation, including a correction of the volume time derivative, slug border mass fluxes, and the local velocity to control volume velocity conversion becomes:

$$
\begin{aligned}
&p_j^{n+1} - p_j^n = \Delta p_j = \\
&+\Delta t \frac{\displaystyle\sum_f \frac{\dot{M}_{f,j}^{src}}{\rho_{f,j}^n} + \Psi_{f,j}^{n+1} - \left(\overline{A}_J^{n+\frac{1}{2}} u_{b,J}^{*n+1} \omega_{b,J} - \overline{A}_{J-1}^{n+\frac{1}{2}} u_{b,J-1}^{*n+1} \omega_{b,J-1} \right) - \sum_f \frac{V_{f,j}^{n+1}}{\rho_{f,j}^n} \left(\frac{\partial \rho_{f,j}}{\partial T} \right)_p \frac{\Delta T_f}{\Delta t}}{\displaystyle\sum_f \frac{V_{f,j}^{n+1}}{\rho_{f,j}^n} \left[\left(\frac{\partial \rho_{f,j}}{\partial p} \right)_{T_f} \right]} \\
&-\Delta t \frac{\displaystyle\sum_f \frac{1}{\rho_{f,j}^n} \left[\begin{array}{l} \hat{m}_{f,J}^{n+1} A_J^n \left(C_{slip,f,J}^n a_J^* u_{k,f,J}^{n+1} + U_{slip,f,J}^n - u_{b,J}^{*n+1} \right) \kappa_{f,J}^{n+1} \\ -\hat{m}_{f,J-1}^{n+1} A_{J-1}^n \left(C_{slip,f,J-1}^n a_{J-1}^* u_{k,f,J-1}^{n+1} + U_{slip,f,J-1}^n - u_{b,J-1}^{*n+1} \right) \kappa_{f,J-1}^{n+1} \end{array} \right]}{\displaystyle\sum_f \frac{V_{f,j}^{n+1}}{\rho_{f,j}^n} \left[\left(\frac{\partial \rho_{f,j}}{\partial p} \right)_{T_f} \right]}
\end{aligned}
\tag{67}
$$

When there are no changes in the cross-sectional area all of the scaling constants becomes 1, and equation (67) reduces to equation (56).

2.9.4 Momentum equation

2.9.4.1 Derivation and discretization

We start by integrating the momentum conservation equation ([39]) for the mixture field *k* over a general momentum control volume (denoted by *j*):

$$\begin{aligned}
&\frac{d}{dt}\int_{V_{k,j}} \rho_{k,j} u_{k,j} dV + \oint_{A_{k,j}} \rho_{k,j}\left(u_{k,j} - u_{b,j}\right) u_{k,j} dA = -\int_{V_{k,j}} \frac{\partial p_j}{\partial x} dV \\
&-\int_{V_{k,j}} \rho_{k,j} g \frac{\partial\left(h_{k,j}\cos\theta\right)}{\partial x} dV - \int_{A_{k,j}} \tau_{i,j} dA - \int_{A_{k,j}} \tau_{k,j,wall} dA - \int_{V_{k,j}} \rho_{k,j} g \sin\theta dV + \int_{V_{k,j}} \dot{\rho}_{k,j}^{src} u_{k,j}^{src} dV
\end{aligned} \tag{68}$$

We include the hydrostatic variation of pressure in the control volume as follows ([39]):

$$\frac{\partial p_{k,j}}{\partial x} = \frac{\partial p_{i,j}}{\partial x} + \rho_{k,j} g \frac{\partial\left(h_{k,j}\cos\theta\right)}{\partial x} \tag{69}$$

Here, $h_{k,j}$ is the fluid height of phase *k* in section *j* and $p_{i,j}$ is the common pressure at the interface (from now on just referred to as p_j).

We can then write:

$$\begin{aligned}
&\frac{d\left(M_{k,j} u_{k,j}\right)}{dt} + \oint_{A_{k,j}} \rho_{k,j}\left(u_{k,j} - u_{b,j}\right) u_{k,j} dA = \\
&-V_{k,j}\frac{\partial p_j}{\partial x} - \int_{V_{k,j}} \rho_{k,j} g \frac{\partial\left(h_{k,j}\cos\theta\right)}{\partial x} dV - \int_{A_{k,j}} \tau_{i,j} dA - \int_{A_{k,j}} \tau_{k,j,wall} dA - \int_{V_{k,j}} \rho_{k,j} g \sin\theta dV + \dot{M}_{k,j}^{src} u_{k,j}^{src}
\end{aligned} \tag{70}$$

For straight sections, not spanning multiple pipes with different inclinations, the inclination does not change along the section length. For bent pipes however, and for sections spanning multiple pipes, the inclination is dependent on the position, and we must evaluate the integrals over $\cos\theta$ and $\sin\theta$ properly. In general, all the integrals over the phase volume in equation

(70) are evaluated as a sum over integrals over the different pipes within the control volume, as described in section 2.9.1.1.

For the gravitational term, we get (assuming in this example that the section spans two pipes, thus being made up of two sub control volumes with lengths L_1 and L_2, and leaving out the gravitational acceleration *g*):

$$\begin{aligned}
&\int_{V_k} \rho_k \sin\theta dV = \int_L \int_{A_k} \rho_k \sin\theta dA dx \\
&= A_k \left(\int_{L_1} \rho_k \sin\theta dx + \int_{L_2} \rho_k \sin\theta dx \right) \\
&= \rho_k A_k \left(\int_{L_1} \sin\theta dx + \int_{L_2} \sin\theta dx \right)
\end{aligned} \tag{71}$$

When θ is not a function of the distance *x*, equation (71) simplifies to:

$$\begin{aligned}
&\rho_k A_k \left[\int_{L_1} \sin\theta dx + A_{k,2} \int_{L_2} \sin\theta dx \right] = \rho_k A_k \left(L_1 + L_2\right) \sin\theta \\
&= \rho_k A_k \left(L_1 + L_2\right) \sin\theta = M_k \sin\theta
\end{aligned} \tag{72}$$

The integral over the sine term for a bent pipe from position x_1 to x_2 is shown in equation (73).

$$\int_{x_1}^{x_2} \sin\theta(x) dx = \frac{R_b}{s} \left[-\cos\left(\theta\left(x_2\right)\right) + \cos\left(\theta\left(x_1\right)\right) \right] \tag{73}$$

The wall shear stress term is also split into a sum of integrals over the different pipe segments:

$$\begin{aligned}
&\int_{A_k} \tau_{k,wall} dA = \int_{L_1} \tau_{k,wall,1} S_{k,1} dx + \int_{L_2} \tau_{k,wall,2} S_{k,2} dx = \tau_{k,wall,1} S_{k,1} L_1 + \tau_{k,wall,2} S_{k,2} L_2 \\
&= \frac{S_{k,wall,1} \lambda_{k,wall,1} \rho_k L_1 \left|u_{k,1}\right| u_{k,1}}{8} + \frac{S_{k,wall,2} \lambda_{k,wall,2} \rho_{k,2} L_2 \left|u_{k,2}\right| u_{k,2}}{8} \\
&= 0.125 \rho_k \left[S_{k,wall,1} \lambda_{k,wall,1} L_1 \left|u_{k,1}\right| u_{k,1} + S_{k,wall,2} \lambda_{k,wall,2} L_2 \left|u_{k,2}\right| u_{k,2} \right]
\end{aligned} \tag{74}$$

This will also take into account different pipe properties like roughness and cross-sectional area in the different sub control volumes. The friction is weighted between the annular and stratified friction models, weighting by $\sin^2(\theta)$. The friction models are described in section 2.10.1.

The weighting of the friction factor multiplied by phase perimeter becomes (note that if any liquid is present, the gas will experience zero wall friction in annular flow, only interfacial friction):

$$\lambda_{k,wall} S_k = \sin^2(\theta)\left(\lambda_{k,wall,annu} S_{k,annu}\right) + \left(1-\sin^2(\theta)\right)\left(\lambda_{k,wall,strat} S_{k,strat}\right) \tag{75}$$

Here, *strat* and *annu* are short for *stratified* and *annular* respectively.

Including this in the integration over the control volume, the integration over a sub control volume in equation (74) becomes:

$$\begin{gathered}\int_{L_1} \tau_{k,1} S_{k,1} dx = \frac{\rho_k \left|u_{k,1}\right| u_{k,1}}{8} \int_{x_1}^{x_2} \lambda_{k,1,wall} S_{k,1} dx = \\ 0.125\rho_k \left|u_{k,1}\right| u_{k,1} \left(\lambda_{k,1,wall,strat} S_{k,1,strat} \int_{x_{1,1}}^{x_{2,1}} \left(1-\sin^2\left(\theta(x)\right)\right) dx + \lambda_{k,1,wall,annu} S_{k,1,annu} \int_{x_{1,1}}^{x_{2,1}} \sin^2\left(\theta(x)\right) dx \right)\end{gathered} \tag{76}$$

Currently, the same wall friction model is used for both stratified and annular flow. The integral over the shear stress in a control volume then becomes:

$$\begin{aligned}\int_{L_j} \tau_{k,wall,j} S_{k,j} dx &= 0.125\rho_{k,j} \sum_s \lambda_{k,wall,j,s} \left|u_{k,j}\right| u_{k,j} \left(W_{strat,j,s} S_{k,strat,j,s} + W_{annu,j,s} S_{k,annu,j,s}\right) \\ &= 0.125\rho_{k,j} \left|u_{k,j}\right| u_{k,j} \sum_s \lambda_{k,wall,j,s} \left(W_{strat,j,s} S_{k,strat,j,s} + W_{annu,j,s} S_{k,annu,j,s}\right)\end{aligned} \tag{77}$$

The summation in equation (77) over the index *s* represents the summation over sub control volumes within a control volume.

The weighting factors W_{strat} and W_{annu} are defined as:

$$\begin{aligned} W_{annu} &= \int_{x_1}^{x_2} \sin^2 \theta(x)\, dx \\ W_{strat} &= \int_{x_1}^{x_2} \left(1 - \sin^2 \theta(x)\right) dx = (x_2 - x_1) - W_{annu} \end{aligned} \tag{78}$$

For a straight pipe the inclination is not a function of position, and we get:

$$\begin{aligned} W_{annu} &= (x_2 - x_1)\sin^2 \theta = L \sin^2 \theta \\ W_{strat} &= L\left(1 - \sin^2 \theta\right) \end{aligned} \tag{79}$$

For a bent pipe, we get:

$$\begin{aligned} W_{annu} &= \left[0.5x - \frac{\sin(2\theta(x))}{4\frac{s}{R_b}} \right]_{x_1}^{x_2} \\ W_{strat} &= (x_2 - x_1) - W_{annu} \end{aligned} \tag{80}$$

Note that for negative inclinations, we use $W_{annu} = 0$ and $W_{strat} = 1$.

Similarly to the wall friction, the interfacial friction factor is also weighted between the stratified and annular models, and summed over the sub control volumes:

$$\begin{aligned} &\int_{A_{k,j}} \tau_{i,j} dA = \\ &0.125 \rho_{g,j} \sum_s \left| u_{k,j,s} - u_{m,j,s} \right| \left(u_{k,j,s} - u_{m,j,s} \right) \left(W_{strat,j,s} \lambda_{i,strat,j,s} S_{i,strat,j,s} + W_{annu,j,s} \lambda_{i,annu,j,s} S_{i,annu,j,s} \right) \end{aligned} \tag{81}$$

Note that the integrated weighting factors W are lengths.

For the interfacial shear stress, different friction factors is used for stratified and annular flow (see section 2.10.1).

The momentum flux term becomes:

$$\int_{A_{k,j}} \rho_{k,j}\left(u_{k,j}-u_{b,j}\right)u_{k,j}dA = m_{f,J}A_J\left(u_{f,J}-u_{b,J}\right)u_{f,J} - m_{f,J-1}A_{J-1}\left(u_{f,J-1}-u_{b,J-1}\right)u_{f,J-1} \tag{82}$$

The momentum flux terms are summed over all of the sub-fields *f* for the mixture field *k*, relating the sub-field velocities to the mixture field by slip relations. If there is only one sub-field, or if no-slip is used between the sub-fields and the mixture field, the result should be the same as using the mixture field values directly (some speedup can however be gained if not summing up all the sub-fields).

The level gradient term is evaluated as a sum of level gradient terms over each static border within a section:

$$\begin{aligned}\rho_k \int_{V_k} \cos\theta(x)\frac{\partial h_k(x)}{\partial x}dV &= \rho_k A_k \int_L \cos\theta(x)\frac{\partial h_k(x)}{\partial x}dx \\ &= \rho_k A_k\left(\int_L \cos\theta_{12}(x)\frac{\partial h_{k,12}(x)}{\partial x}dx + \int_L \cos\theta_{23}(x)\frac{\partial h_{k,23}(x)}{\partial x}dx\right)+\ldots\end{aligned} \tag{83}$$

Here sub-script 12 means between pipe nr. 1 and pipe nr. 2, and so forth. To account for variations in cross sectional pipe area, each individual level gradient term in equation (83) is evaluated as (here showing the level gradient between pipe 1 and pipe 2):

$$\int_{x_1}^{x_2} \cos\theta(x)\frac{\partial h_k(x)}{\partial x}dx = \left(h_{k,2}+\Delta z_2\right)\cos\theta\left(x_2\right) - \left(h_{k,1}+\Delta z_1\right)\cos\theta\left(x_1\right) \tag{84}$$

In equation (84) a possible difference in the pipe diameter between the two pipes has been included to correct the level gradient, in the term $\Delta z_i = \left(0.5D_{i-1} - 0.5D_i + \Delta h_{i-1,i}\right)$. The variable

$\Delta h_{i-1,i}$ in this term is a possible user-specified height offset between the centres of the two pipes. This offset can have values in the range:

$$-\left|\frac{D_1}{2}+\frac{D_2}{2}\right| \le \Delta h_{1,2} \le \left|\frac{D_1}{2}+\frac{D_2}{2}\right| \tag{85}$$

The default is $\Delta h_{1,2} = 0$, so that the pipes are connected at the pipe centres.

Figure 27 and Figure 28 shows a test case with a change in the pipe diameter (0.1 and 0.2 m), to test the implementation of the level gradient term. The first test case use 128 straight pipe segments to form an arc starting at -80 degrees and ending at +80 degrees, while the second test case utilize the bent pipe framework to create the same geometry using only 2 pipes. The total length is 2 m, and a constant time step of 0.01 s is used. Three different offsets between the pipe centres at the location of the area change were used: 0.05, 0, and -0.05 m. This corresponds to the pipes being connected at the bottom, centre and top respectively. The initial condition is some liquid distributed in the middle of the pipe, but not with a horizontal interface. It can be seen from Figure 27 and Figure 28 that the proposed level gradient will make the gas and liquid level stabilize with a horizontal and flat interface in all cases, as expected.

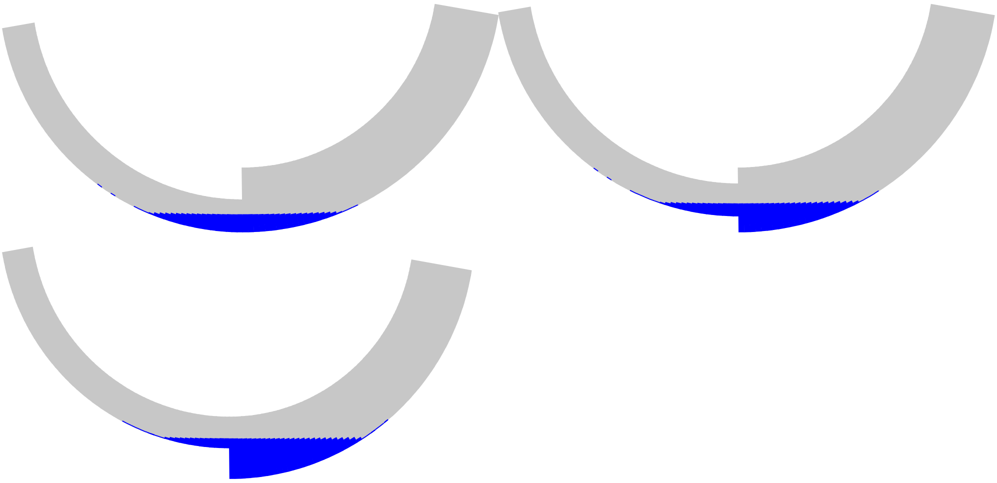


Figure 27: Fluids at rest in a system with 128 sections and 128 straight pipes, with different pipe diameters in the left and right parts of the system.

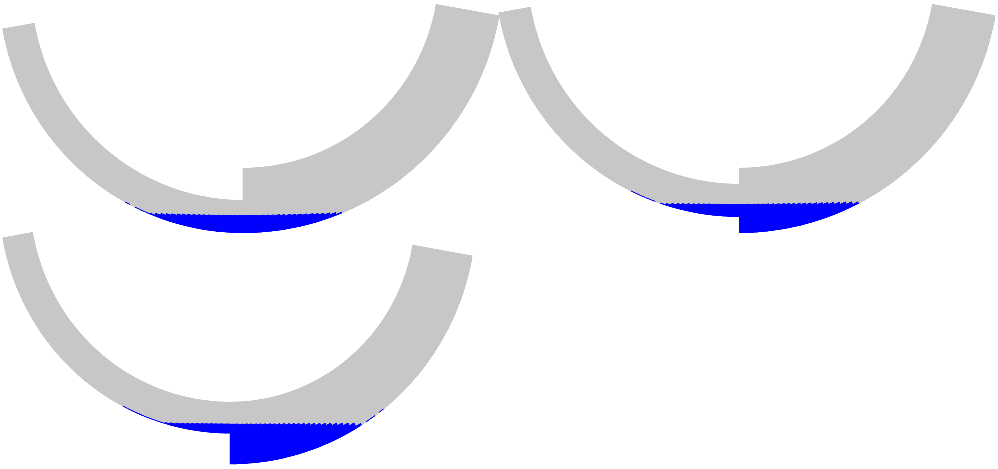


Figure 28: Fluids at rest in a system with 128 sections and two bent pipe segments, with different pipe diameters in the left and right parts of the system.

As an alternative to summing up the integrated friction, gravity and level gradient terms over the different pipe segments, one could use average values. For instance for a border separating two straight pipes with two different inclinations θ_1 and θ_2, the corresponding gravitational term for the integration approach and the average values approach would become as shown in equations (86) and (87) respectively:

$$F_{g,int} = M_k g \frac{1}{L_1 + L_2}\left(L_1 \sin\left(\theta_1\right) + L_2 \sin\left(\theta_2\right)\right) \tag{86}$$

$$F_{g,avg} = M_k g \sin\left(0.5\left(\theta_1 + \theta_2\right)\right) \tag{87}$$

Similarly one could use the average pipe properties, holdups and phase perimeters in the friction terms, and average inclination for the cosine term in the level gradient. A comparison between these two methods have been performed, using a pipe geometry with first a negative inclination of -3 degrees and then a positive inclination of 60 degrees. Both pipes have an equal diameter of 0.05 m, and lengths of 4 m. The initial liquid holdup is set to 0.02 everywhere. No flow rates are specified, so that after the simulation starts the holdup will start accumulating in the low point. Three different test cases were run: One with a constant grid size of 0.125 m, one where the grid size alternates between 0.1 and 0.3 m, and one where the grid size alternates between 0.1 m and 0.9 m. The simulations were run for 200 s, and the results are shown in Figure 29 - Figure 35.

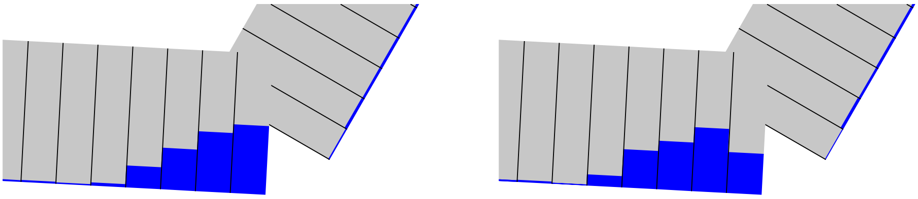


Figure 29: Liquid holdup at time 200 s for the test case with a constant grid size. The left image shows the result for the integration approach, while the right result is for using average values. The pipe diameter is scaled by a factor of 10 for better illustration.

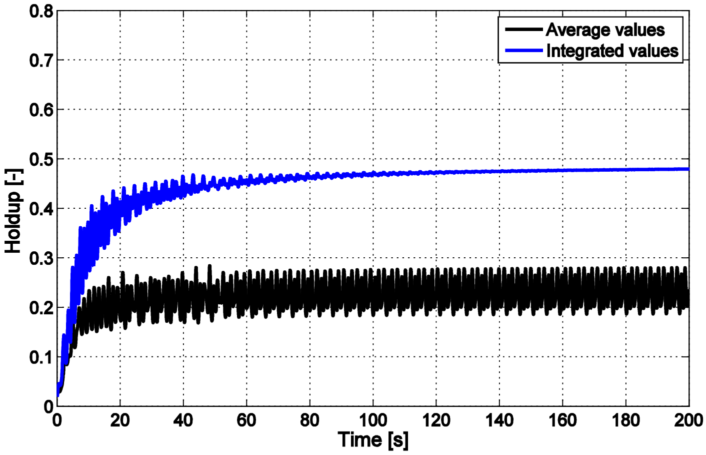


Figure 30: Liquid holdup in the last section in the negative inclined pipe plotted against time for a test case with liquid accumulating in a low point. The grid size is approximately 0.14 m.

Figure 29 shows that the result obtained with the integration approach looks more physical than the result from using average values, with a more linear slope of the liquid level in the low point. Furthermore, it can be seen from Figure 30 that the liquid in the low point oscillates and never comes to rest when using the average values approach, while the integration approach stabilizes at a constant value. The results from the simulation using a grid size of 0.1 m and 0.3 m are show in Figure 31 - Figure 33.

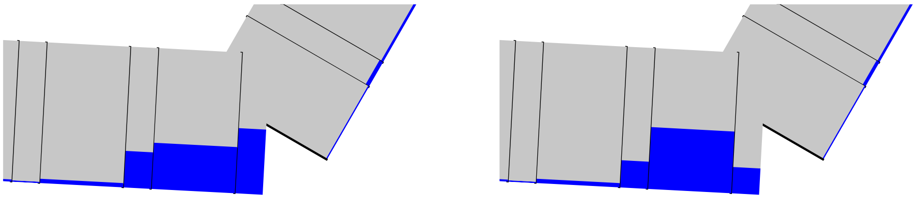


Figure 31: Oscillating liquid holdup at two different time levels for the test case with an alternating grid size of 0.1 and 0.3 m. Both images shows the approach using average values. The pipe diameter is scaled by a factor of 10 for better illustration.

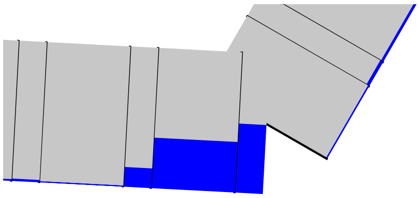


Figure 32: Liquid holdup at time 200 s for the test case with an alternating grid size of 0.1 and 0.3 m for the integration approach. The pipe diameter is scaled by a factor of 10 for better illustration.

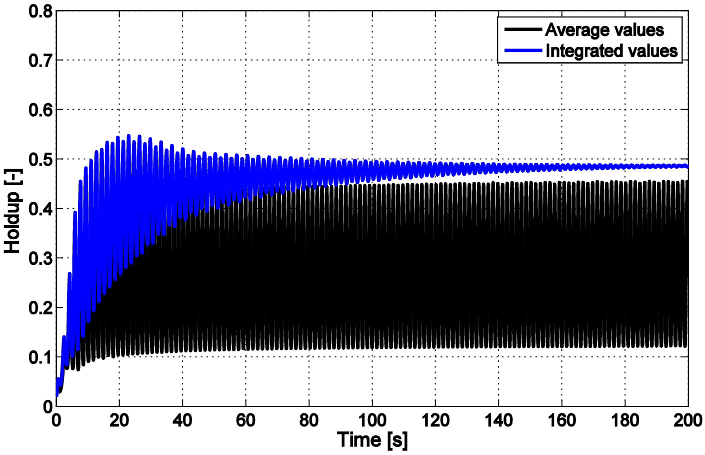


Figure 33: Liquid holdup in the last section in the negative inclined pipe plotted against time for a test case with liquid accumulating in a low point. The grid size is alternating between 0.1 and 0.3 m.

Figure 31 - Figure 33 shows that the results for the test case with an alternating grid size of 0.1 m and 0.3 m are similar to the test case with a constant grid size, with a more physical looking linear slope of the liquid level in the low point for the integration approach. For this test case however the amplitude of the oscillations in the holdup from using the average values approach is much larger than in the test case with a constant grid size.

Similar results are seen in the final test case, with the largest grid size ratio (0.1 and 0.9 m), shown in Figure 34 - Figure 36. The liquid level stabilizes for the integration approach, while oscillations are seen with the average approach with an even larger amplitude than in the previous test case.

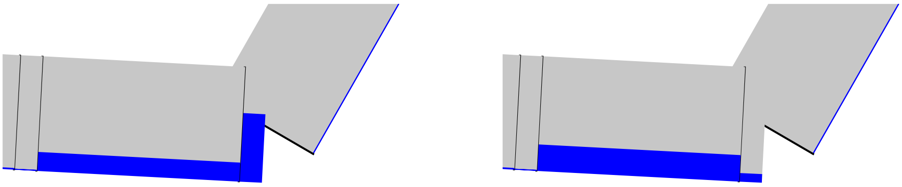


Figure 34: Oscillating liquid holdup at two different time levels for the test case with an alternating grid size of 0.1 and 0.9 m. Both images shows the approach using average values. The pipe diameter is scaled by a factor of 10 for better illustration.

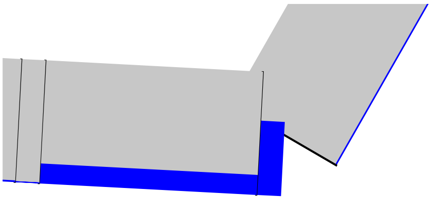


Figure 35: Liquid holdup at time 200 s for the test case with an alternating grid size of 0.1 and 0.9 m for the integration approach. The pipe diameter is scaled by a factor of 10 for better illustration.

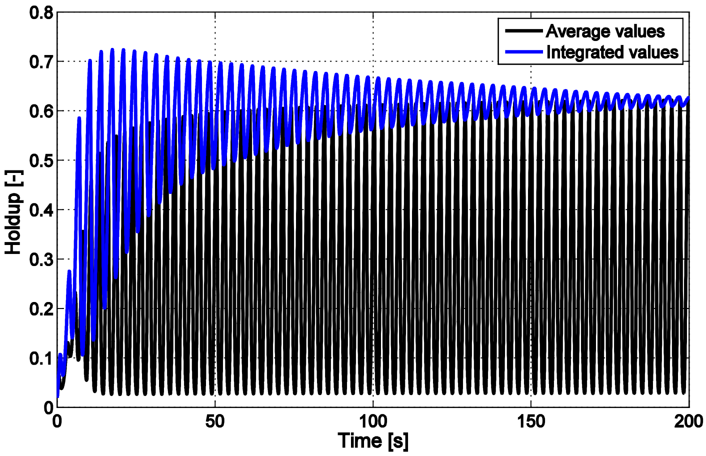


Figure 36: Liquid holdup in the last section in the negative inclined pipe plotted against time for a test case with liquid accumulating in a low point. The grid size is alternating between 0.1 and 0.9 m.

A more detailed inspection of the presented test cases showed that it is the wall friction term that makes the biggest difference between the integration approach and the approach using average values. The biggest difference was not surprisingly found at the static border separating the pipes with different inclinations. The reason for this is that there will be a very small holdup value in the right half control volume of the static border. When integrating up the friction from the different parts, the right half has a very low holdup and will obtain a quite small hydraulic diameter and Reynolds number. This will lead to a relatively large laminar friction factor, which makes the total liquid wall friction for the static border relatively large. If using average values however the low liquid holdup value in the right half will not be used, and one will get a much smaller friction because the average holdup is used. This might sound somewhat counter-intuitive, but seems to lead to a more physically correct result. Both approaches however use a quite long time before the oscillations die out, but this is a well-known problem in the two fluid model. The friction models are defined for steady state flow, and gives a too small value for this highly dynamic test case. This issue is discussed for instance in [40], where dynamic friction models are tested for simulations of oscillating liquid in a u-tube geometry.

From reasoning one can also see that the integrated gravitational term is probably better than just using the average inclination: If the length of the right half of the control volume goes to zero ($L_2 \to 0$ in equations (86) and (87)) the gravitational term using the average inclination in equation (87) remains the same, while the integrated gravity term in equation (86) goes towards $M_k g \sin(\theta_1)$.

The final (semi-implicit) discretized version of the momentum equation becomes:

$$
\begin{aligned}
&\frac{M_{k,j}^{n+1}u_{k,j}^{n+1}-M_{k,j}^{n}u_{k,j}^{n}}{\Delta t}\\
&+\sum_{f}\left[m_{f,J}^{n+1}A_{J}^{n}\left(u_{f,J}^{n+1}-u_{b,J}^{n+1}\right)\hat{u}_{f,J}^{n+1}-m_{f,J-1}^{n+1}A_{J-1}^{n}\left(u_{f,J-1}^{n+1}-u_{b,J-1}^{n+1}\right)\hat{u}_{f,J-1}^{n+1}\right]\\
&=-V_{k,j}^{n}\frac{\left(p_{J}^{n+1}-p_{J-1}^{n+1}\right)}{L_{j}}+\sum_{s}\dot{M}_{k,j,s}^{src,n}u_{k,j,s}^{src,n}\\
&-0.125\rho_{k,j}^{n}\left|u_{k,j}^{n}\right|u_{k,j}^{n+1}\sum_{s}\lambda_{k,wall,j,s}^{n}\left(W_{strat,j,s}^{n}S_{k,strat,j,s}^{n}+W_{annu,j,s}^{n}S_{k,annu,j,s}^{n}\right)\\
&-0.125\rho_{g,j}^{n}\sum_{s}\left|u_{k,j}^{n}-u_{m,j}^{n}\right|\left(u_{k,j}^{n+1}-u_{m,j}^{n+1}\right)\left(W_{strat,j,s}^{n}\lambda_{i,strat,j,s}^{n}S_{i,strat,j,s}^{n}+W_{annu,j,s}^{n}\lambda_{i,annu,j,s}^{n}S_{i,annu,j,s}^{n}\right)\\
&-\rho_{k,j}^{n}gA_{k,j}\sum_{s}\left(h_{k,J,s}\cos\theta_{J,s}-h_{k,J-1,s}\cos\theta_{J-1,s}\right)-M_{k,j}^{n}g\frac{1}{L_{j}^{n}}\int_{L_{j}^{n}}\sin\left(\theta_{j}\left(x\right)\right)dx\\
&\pm\phi_{e,j}^{n}u_{e,j}^{n+1}\pm u_{d,j}^{n+1}\phi_{d,j}^{n}+\sum_{s}F_{pert,s}+\sum_{s}F_{valve,s}+\sum_{s}F_{\mathrm{minorlosses},s}+V_{k,j}\sum_{s}\frac{\partial \mathrm{P}_{userforce,s}}{\partial x}
\end{aligned}
\tag{88}
$$

The BDF2 and Crank-Nicolson time integration methods can also be used for the momentum equation, see equation (29).

Here the effect on momentum from all types of entrainment and deposition rates has been added, and the effect from additional losses like valves, contraction and expansion losses, perturbations and user-defined forces (see Appendix C.2 - Appendix C.4). The velocity of the field the entrainment is taken from is named $u_{e,j}^{n+1}$, while the velocity of the field the deposition rate comes from is named $u_{d,j}^{n+1}$. Note that the sign of the entrainment and deposition terms will be different in the gas and liquid momentum equations. For instance, oil droplets in the gas will have a positive sign in the mixture gas momentum equation, and a negative sign in the mixture liquid momentum equation. For gas bubbles in oil, the sign is positive for mixture liquid and negative for mixture gas. In general, if the entrained sub-field belongs to the mixture field for the equation we look at, the sign is positive.

By splitting the new velocities into the old value plus a difference, the momentum equation gives a banded matrix for the change in the gas and liquid velocities at the borders. Note that though the momentum equation is solved for velocity, it is still formulated as a conservation equation for momentum.

The mass flux terms containing the new masses, volumes and velocities in equation (88) are not split up into an old value plus a change term. These terms are instead evaluated using the new values for mass, volume and velocity, which are kept and updated in each iteration on the system of equations. These terms will thus be converging to the correct final value in each iteration. Newton's method can alternatively be used to solve the system of equations. The solution procedure is described in more detail in section 2.14.1.

It is also possible to use a fully implicit momentum equation. However, this will result in a much more complicated analytical Jacobian, which is not implemented. The new values for each non-linear term in the momentum equation will instead be used in each new iteration, converging towards a fully implicit solution. This is not the ideal way to solve a fully implicit system, and one might encounter convergence issues. It is however also possible to calculate the Jacobian numerically, though this will typically be much slower.

2.9.4.2 Explicit momentum equation

To specify the velocity for a momentum control volume explicitly, the coefficients in the momentum equation are updated as follows:

$$\begin{aligned} c_{k,diag} &= 1 \\ rhs_k &= \left(u_k^{n+1} - u_k^n\right) \end{aligned} \tag{89}$$

Here $c_{k,diag}$ is the diagonal coefficient, while rhs_k is the right hand side.

This will yield a solution $\Delta u_k = \dfrac{rhs_k}{c_{k,diag}} = \left(u_k^{n+1} - u_k^n\right)$.

This is done for the mass flow boundaries, and for incompressible plugs (the entire incompressible plug unit) that are in contact with a mass flow boundary. The mass flow boundaries are described in section 2.9.6.2.

2.9.4.3 Section velocities

Section velocities are updated after the velocities for the borders have been solved, and are calculated as the average between the neighbouring border velocities. For incompressible plug sections however, the velocity is calculated from the volume flow in the plug unit using the local cross sectional area, as described in section 2.12.4.

2.9.4.4 Area changes

To take into account variations in the cross-sectional area, let us have a look at the first term in equation (68), since a control volume might span pipes with different properties.
By carrying out the integration over the sub control volumes (denoted by *s*) for control volume *j*, we get:

$$\frac{d}{dt}\int_{V_{k,j}} \rho_{k,j} u_{k,j} dV = \frac{d}{dt}\left(\rho_{k,j}\int_{V_{k,j}} u_{k,j} dV\right) = \frac{d}{dt}\left(\rho_{k,j}\sum_{s} A_{k,j,s} u_{k,j,s} L_{j,s}\right) \tag{90}$$

As an example to illustrate the use of the sub control volume indexing, *s*=2 and *j*=1 would mean sub control volume number 2 in control volume number 1.

As described in section 2.9.1.3 in equation (45), we approximate the local velocity to be equal to the control volume velocity scaled by the average control volume area to the local area.

We then get:

$$\begin{aligned}&\frac{d}{dt}\left(\rho_{k,j}\sum_{s}\left[A_{k,j,s}u_{k,j,s}L_{j,s}\right]\right)=\frac{d}{dt}\left(\rho_{k,j}\sum_{s}\left[A_{k,j,s}u_{k,j}\frac{\overline{A}_{k}}{A_{k,j,s}}L_{j,s}\right]\right)=\\&\frac{d}{dt}\left(\rho_{k,j}u_{k,j}\overline{A}_{k,j}\sum_{s}\left[L_{j,s}\right]\right)=\frac{d}{dt}\left(\rho_{k,j}u_{k,j}\overline{A}_{k,j}L_{j}\right)=\frac{d}{dt}\left(\rho_{k,j}u_{k,j}V_{k,j}\right)=\frac{d}{dt}\left(M_{k,j}u_{k,j}\right)\end{aligned} \tag{91}$$

This is the same result as when not taking into account changes in area.

The velocities used when summing up the wall and interfacial friction terms in the momentum equation are all scaled by the area ratio factor a^{*} (see equation (46)), so that a part of the control volume with a smaller cross sectional area will get a larger velocity and higher friction.

The other type of velocity scaling is related to the upwind velocities. Instead of calculating the upwind velocity at the locations where the new velocities are not defined, the upwind volume flow is calculated instead. This is also done for the higher order upwind schemes/flux limiters. The velocity can then be calculated from the upwind volume flow, by dividing by the cross-sectional area at the location where we need the velocity. This was found to give the correct pressure profile in a single phase incompressible test case without friction, a case which can be calculated using the Bernoulli equation [41]:

$$\frac{1}{2}\rho u^{2}+\rho gh+p=const \tag{92}$$

As described in section 2.9.3.1.2 regarding the pressure equation, scaling factors to convert from local velocity to control volume velocity in the pressure equation are also necessary to include to obtain the correct results. This will be illustrated in the following plots.

In the case of a horizontal pipe the gravity term drops out of the Bernoulli equation, and the pressure at location A2 in Figure 37 can be calculated as:

$$
\begin{aligned}
& p_2 = p_1 + \frac{1}{2}\rho\left(u_1^2 - u_2^2\right) = \\
& p_2 = p_1 + \frac{1}{2}\rho\left(u_1^2 - \left(\frac{u_1 A_1}{A_2}\right)^2\right) = \\
& p_1 + \frac{1}{2}\rho u_1^2\left(1 - \frac{A_1^2}{A_2^2}\right)
\end{aligned}
\tag{93}
$$

It the following calculations it will be shown that using the upwind volume flow and then dividing by the local area gives the correct Bernoulli pressure effect in the case of incompressible flow without friction. The following derivation is for the default discretization in Sluggit, where the momentum fluxes consists of one velocity from a central difference multiplied by one velocity from a general flux limiter.

In the following derivation we use the first order upwind scheme, and assume flow from left to right. A simulation test case is also presented, showing that the proposed scaling gives the desired pressure field. The test case geometry is shown in Figure 37, and consists of single phase flow with nearly incompressible gas (compressibility 1e-8 kg/Pa) with a density of 1 kg/m3, and with a velocity of 1 m/s at the inlet border. The pipe diameter in the large and small part of the pipe is 0.1 and 0.025 m respectively, which gives $A_1 = 16A_2$. Since the flow is nearly incompressible, this also gives $u_1 \approx 16u_2$. A constant pressure boundary of 101300 Pa is used at the outlet. No additional losses due to expansion/contraction is enabled.

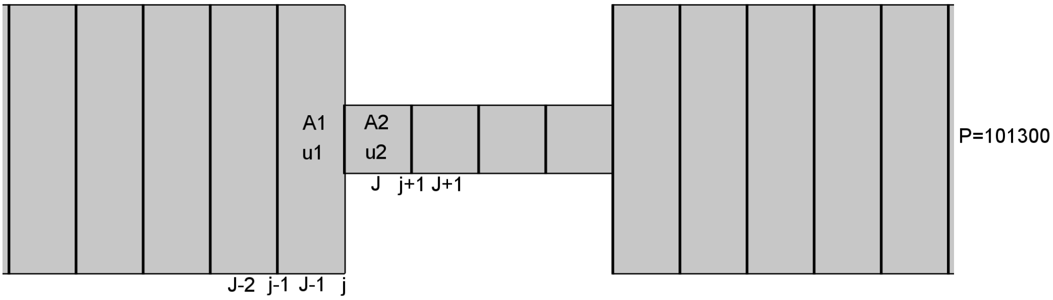


Figure 37: Frictionless horizontal test case with an area reduction by a factor of 16.

The result of the simulation is shown below in Figure 38.

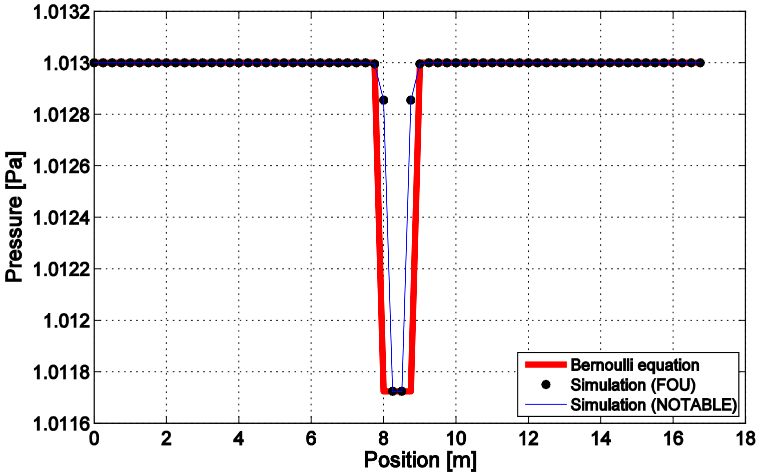


Figure 38: Pressure field for incompressible gas flow in a geometry with changes in cross-sectional area, using the FOU and NOTABLE convection schemes.

It can be seen that the pressure field exactly matches the Bernoulli equation, except for in the sections in contact with the area change (sections indicated by *J*-1 and *J* in Figure 37) where there is an entrance effect. In the other sections with the smaller cross sectional area, the pressure field is correct. The results from using the third order NOTABLE scheme is also included, which gives the same result as the first order upwind scheme. This indicates that the proposed scaling of the upwind velocities can also be used with higher order schemes. It will now be derived analytically how these results are obtained, for the first order scheme (a similar derivation can be made for higher order upwind schemes). For the static border between A_1 and A_2 denoted by index *j*, the momentum equation for incompressible flow without losses becomes (this expression is general for all borders):

$$
\begin{aligned}
&\rho A_J u_J \hat{u}_J - \rho A_{J-1} u_{J-1} \hat{u}_{J-1} = -V_j \frac{\left(p_J - p_{J-1}\right)}{\Delta x_j} = -A_{j,avg}\left(p_J - p_{J-1}\right) \\
&\rho A_J u_J \frac{u_j A_j}{A_J} - \rho A_{J-1} u_{J-1} \frac{u_{j-1} A_{j-1}}{A_{J-1}} = -A_{j,avg}\left(p_J - p_{J-1}\right) \\
&\rho A_J u_J \frac{Q}{A_J} - \rho A_1 u_{J-1} \frac{Q}{A_1} = -A_{j,avg}\left(p_J - p_{J-1}\right) \\
&\rho Q u_J - \rho Q u_{J-1} = -A_{j,avg}\left(p_J - p_{J-1}\right) \\
&p_J = p_{J-1} + \frac{1}{A_{j,avg}}\left(\rho Q u_{J-1} - \rho Q u_J\right)
\end{aligned}
\tag{94}
$$

Here we have replaced the upwind velocities by the velocities in the left control volume, assuming flow from left to right. The average area of the control volume for the border at index *j* is named $A_{j,avg}$, and becomes: $A_{j,avg} = 0.5\left(A_1 + A_2\right) = 0.5\left(16A_2 + A_2\right) = 8.5A_2$.
Equation (94) will be used to calculate expressions for the pressure at indexes from *J*-2 to *J*+1, showing that the total pressure drop matches that given by the Bernoulli equation.
In equation (94) it has been used that the volume flow *Q* is constant throughout the domain, and inserted the scaling of the upwind volume flow divided by local area for the upwind velocity.

For the border *before* the static border at the area change (index *j*-1) we get:

$$\begin{aligned}
&A_{j-1,avg}\left(p_{J-1}-p_{J-2}\right)=\rho Q u_{J-2}-\rho Q u_{J-1}\\
&A_1\left(p_{J-1}-p_{J-2}\right)=\frac{\rho}{2}Q\left(u_1+u_1\right)-\frac{\rho}{2}Q\left(u_1+u_j\right)\\
&A_1\left(p_{J-1}-p_{J-2}\right)=\frac{\rho}{2}Qu_1-\frac{\rho}{2}Qu_j\\
&A_1\left(p_{J-1}-p_{J-2}\right)=\frac{\rho}{2}A_1u_1^2-\frac{\rho}{2}A_1u_1\frac{A_1u_1}{A_{j,avg}}\\
&A_1\left(p_{J-1}-p_{J-2}\right)=\frac{\rho}{2}u_1^2A_1\left(1-\frac{A_1}{A_{j,avg}}\right)\\
&p_{J-1}=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1}{A_{j,avg}}\right)
\end{aligned}\tag{95}$$

This expression gives a pressure drop of 0.4416 Pa, which is exactly what is obtained in the simulation.

For the static border between A_1 and A_2, at index *j*, we get:

$$\begin{aligned}
&A_{j,avg}\left(p_J-p_{J-1}\right)=\rho Q u_{J-1}-\rho Q u_J\\
&A_{j,avg}\left(p_J-p_{J-1}\right)=\frac{\rho}{2}Q\left(u_1+u_j\right)-\frac{\rho}{2}Q\left(u_j+u_2\right)\\
&A_{j,avg}\left(p_J-p_{J-1}\right)=\frac{\rho}{2}Qu_1-\frac{\rho}{2}Qu_2\\
&A_{j,avg}\left(p_J-p_{J-1}\right)=\frac{\rho}{2}A_1u_1^2-\frac{\rho}{2}A_1u_1u_2\\
&p_J=p_{J-1}+\frac{\rho}{2}u_1^2\frac{A_1}{A_{j,avg}}\left(1-\frac{A_1}{A_2}\right)
\end{aligned}\tag{96}$$

This gives a pressure drop of 14.1319 Pa, also consistent with the simulation.

For the border to the right of the static border, at index *j*+1, we get:

$$\begin{aligned}
&A_{j+1,avg}\left(p_{J+1}-p_J\right)=\rho Q u_J-\rho Q u_{J+1} \\
&A_2\left(p_{J+1}-p_J\right)=\frac{\rho}{2}Q\left(u_j+u_2\right)-\frac{\rho}{2}Q\left(u_2+u_2\right) \\
&A_2\left(p_{J+1}-p_J\right)=\frac{\rho}{2}A_1u_1u_j-\frac{\rho}{2}u_1A_1u_2 \\
&A_2\left(p_{J+1}-p_J\right)=\frac{\rho}{2}A_1u_1\left(\frac{A_1u_1}{A_{j,avg}}\right)-\frac{\rho}{2}u_1A_1\left(\frac{A_1u_1}{A_2}\right) \\
&p_{J+1}=p_J+\frac{\rho}{2}u_1^2\frac{A_1}{A_2}\left(\frac{A_1}{A_{j,avg}}-\frac{A_1}{A_2}\right)
\end{aligned} \tag{97}$$

Combining equation (95) and (96), we get for the total pressure drop from index *J*-2 to *J*:

$$\begin{aligned}
&p_J=p_{J-1}+\frac{\rho}{2}u_1^2\frac{A_1}{A_{j,avg}}\left(1-\frac{A_1}{A_2}\right) \\
&p_J=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1}{A_{j,avg}}\right)+\frac{\rho}{2}u_1^2\frac{A_1}{A_{j,avg}}\left(1-\frac{A_1}{A_2}\right) \\
&p_J=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1}{A_{j,avg}}+\frac{A_1}{A_{j,avg}}\left(1-\frac{A_1}{A_2}\right)\right) \\
&p_J=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1^2}{A_{j,avg}A_2}\right)
\end{aligned} \tag{98}$$

Combining equation (98) and (97), the total pressure drop from index *J*-2 to *J*+1 becomes:

$$\begin{aligned}
&p_{J+1}=p_J+\frac{\rho}{2}u_1^2\frac{A_1}{A_2}\left(\frac{A_1}{A_{j,avg}}-\frac{A_1}{A_2}\right) \\
&p_{J+1}=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1^2}{A_{j,avg}A_2}\right)+\frac{\rho}{2}u_1^2\frac{A_1}{A_2}\left(\frac{A_1}{A_{j,avg}}-\frac{A_1}{A_2}\right) \\
&p_{J+1}=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1^2}{A_{j,avg}A_2}+\frac{A_1}{A_2}\left(\frac{A_1}{A_{j,avg}}-\frac{A_1}{A_2}\right)\right) \\
&p_{J+1}=p_{J-2}+\frac{\rho}{2}u_1^2\left(1-\frac{A_1^2}{A_2^2}\right)=p_{J-2}+\frac{\rho}{2}\left(u_1^2-u_2^2\right)
\end{aligned} \tag{99}$$

This expression proves that the total pressure drop from the section in Figure 37 marked by index *J*-2 to the section marked by index *J*+1 equals exactly that of the Bernoulli equation. The proposed scaling only consists of area ratios, which means that there will be no modifications when there are no changes in the cross-sectional pipe area. The derivation also shows that to obtain the correct Bernoulli pressure drop, the region with the smaller area must be split up into a minimum of three sections, as there will be an inlet effect in the first section within the region with an area change. Note also that in this derivation it is assumed for the momentum fluxes that one velocity is from a general flux limiter, while the other velocity is a central difference. If the square value of the upwind velocity is used instead, equations (95) - (100) does not apply and other modifications of the momentum flux would be necessary to obtain the correct pressure field.

Simulations without the previously discussed scaling factors applied were also run, and the results are shown Figure 39.

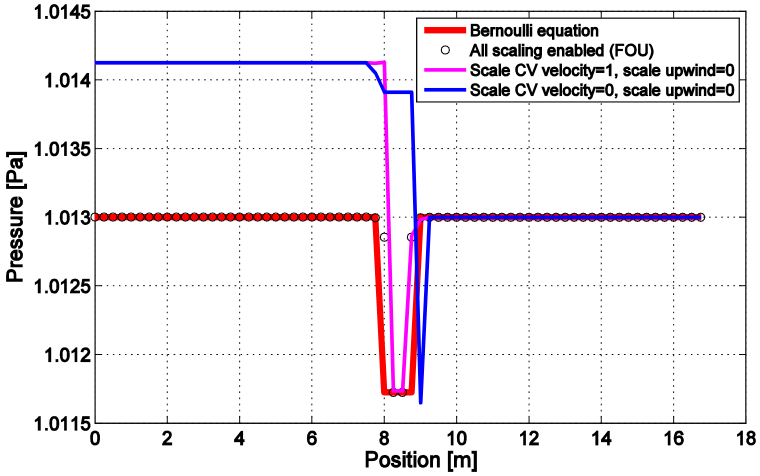


Figure 39: Pressure field for incompressible gas flow in a geometry with changes in cross-sectional area, with different settings for the area scaling factors in the simulations.

From Figure 39 it can be seen that turning off the upwind area scaling factors gives the correct result from the outlet and until the middle of the constriction, whereupon the pressure field deviates from the analytical result to a too large value. For the case with both the upwind scaling and the local velocity scaling in the pressure equation turned off, the result is even worse. The pressure field is correct from the outlet pressure boundary and until the location with a change in cross sectional area, but from this point the pressure field deviates from the Bernoulli equation. In opposite to only disabling the upwind scaling factors, the pressure inside the constriction is now wrong.

The same simulations as shown in Figure 39 were also run with a compressible gas, and with the friction enabled. The Bernoulli equation for compressible flow reads:

$$\frac{1}{2}\rho_1 u_1^2 + \rho_1 g h_1 + p_1 = \frac{1}{2}\rho_2 u_2^2 + \rho_2 g h_2 + p_2 + h_{loss} \tag{100}$$

Here, h_{loss} is the head loss between points 1 and 2. The same method as for incompressible flow was used in the compressible simulations, which will be an ok assumption as long as the head loss in equation (101) is relatively small compared to the other terms.

In these simulations the grid was refined to get more points when plotting to make it easier to visualize the pressure field inside the constriction, and the compressibility was set to 1e-5 kg/Pa. These results are presented in Figure 40.

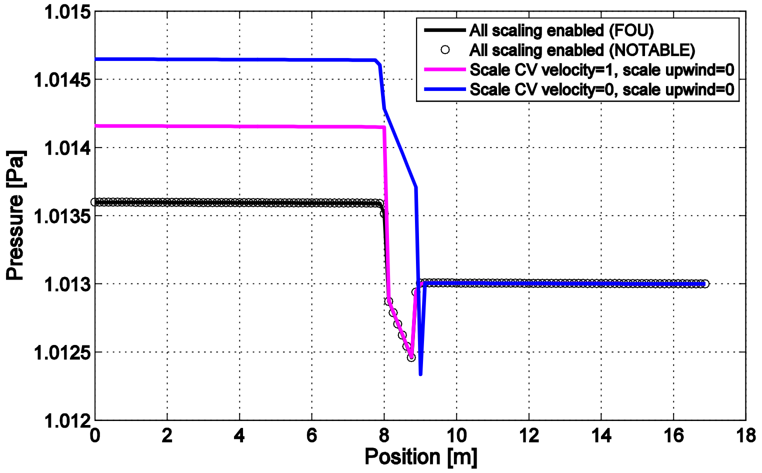


Figure 40: Pressure field for compressible gas flow in a geometry with changes in cross-sectional area, with different settings for the area scaling factors in the simulations.

It can be seen from Figure 40 that the first order and third order upwind schemes with all area scaling factors enabled give an identical result also in the case of compressible flow. Disabling the scaling factors gives the wrong pressure field, just as for the incompressible case. In opposite to the incompressible case, the two different scaling settings now give a different pressure in the start of the pipe. It can also be seen that the effect from disabling the scaling of the local velocities in the pressure equation seems to give a larger error than disabling the scaling of the upwind velocity.

The proposed scaling of the upwind velocities are as mentioned derived for the case of one central difference multiplied by a general flux limiter value. In the case of using the square of the flux limiter value, the scaling needs to be different.

2.9.5 Energy equation

The energy equation is solved for the mixture gas and mixture liquid, and for the pipe wall segments. The energy equation is solved for the change in specific enthalpy, which is stored in the cell centres. The temperature can then be calculated from the heat capacity and the new specific enthalpy for an ideal fluid, or from finding the temperature that corresponds to the enthalpy in the PVT-file.

We start by integrating the energy equation over the control volume V_k for (phase *k*):

$$\frac{d}{dt}\left[M_k\left(h_k+\frac{u_k^2}{2}\right)\right]+\int_{A_k}\rho_k\left(u_k-u_b\right)\left(h_k+\frac{u_k^2}{2}\right)dA=$$

$$\frac{d\left(\langle p\rangle V_k\right)}{dt}-\int_{A_k}p_k u_b dA+\int_{A_k}\tau_k u_k dA+\int_{A_k}k\nabla T dA+\dot{M}_k^{src}\left(h_k^{src}+\frac{\left(u_k^{src}\right)^2}{2}\right)-M_k u_k g\frac{1}{L}\int_L \sin\left(\theta\left(x\right)\right)dx \tag{101}$$

Here, *h* is the specific enthalpy, *k* is the conductivity, *T* is the temperature, while h^s and u^s are the specific enthalpy and velocity of a fluid mass source. The heat transfer terms to for instance the other fluid and to the pipe walls are introduced as a sum of the different contributions in the final form of the energy equation, which is shown in equation (109).

The pressure integral is (to first order):

$$\begin{aligned}&\int_{A_k}p_k u_b dA=\langle p\rangle\int_{A_k}u_b dA+\int_{A_k}\left(p_k-\langle p\rangle\right)u_b dA\approx\\&\langle p\rangle\frac{dV_k}{dt}+V_k\langle u_b\rangle\frac{\partial\langle p\rangle}{\partial x}\end{aligned} \tag{102}$$

Here, the values in brackets $\langle\ \rangle$ are averaged values.

Removing the averaging brackets to simplify, the energy equation can be written:

$$\begin{aligned}&\frac{d}{dt}\left(M_k\left(h_k+\frac{u_k^2}{2}\right)\right)+\int_{A_k}\rho_k\left(u_k-u_b\right)\left(h_k+\frac{u_k^2}{2}\right)dA=\\&V_k\left(\frac{dp}{dt}-u_b\frac{\partial p}{\partial x}\right)+\int_{A_k}\tau_k u_k dA+\int_{A_k}k\nabla T dA+\dot{M}_k^{src}\left(h_k^{src}+\frac{\left(u_k^{src}\right)^2}{2}\right)-M_k u_k g\frac{1}{L}\int_L\sin\left(\theta\left(x\right)\right)dx\end{aligned}\tag{103}$$

The wall shear stress should not be included as it does not perform any work (velocity is zero at surface of the control volume, at the wall), it is only the interfacial shear stress that performs work. Following the same integration procedure for the interfacial friction as shown section 2.9.4.1, we get:

$$\int_{A_k}\tau_k u_k dA=0.125\rho_{g,j}^{n+1}\sum_s\left|u_{k,j}^{n+1}-u_{m,j}^{n+1}\right|\left(u_{k,j}^{n+1}-u_{m,j}^{n+1}\right)W_{j,s}u_{l,j}^{n+1}\tag{104}$$

Here the interfacial friction term has been simplified to make the expression shorter, by using:

$$W_{j,s}=\left(W_{strat,j,s}\lambda_{i,strat,j,s}^{n}S_{i,strat,j,s}^{n}+W_{annu,j,s}\lambda_{i,annu,j,s}^{n}S_{i,annu,j,s}^{n}\right)\tag{105}$$

It is also assumed that the interface velocity is equal to the liquid velocity ($u_{i,j}^{n+1}=u_{l,j}^{n+1}$).

Just as is done in the momentum equation, the convective terms are summed over all of the sub-fields *f* for the mixture field *k*, relating the sub-field velocities to the mixture field by slip relations. It is also implemented to not sum up the sub-fields and just use the mixture field directly. Note that if a slip relation other than no-slip is used for instance for the droplet field of for the gas bubbles in liquid, it will be incorrect to not sum up the sub-fields as different fields can move with different velocities. In summing up the contribution from the individual sub-fields, the energy equation can then no longer be solved directly as a linear system of equations for the change in the specific mixture enthalpy. This is because there is no simple way of relating the specific sub-field enthalpy to the specific mixture field enthalpy in a one-to-one relation (which in opposite is possible with the sub-field velocities in the momentum equation). The specific mixture enthalpy is related to all sub-fields, as shown in equation (107).

$$M_{k,j}h_{k,j} = \sum_{f} M_{f,j}h_{f,j} \tag{106}$$

To solve the energy equation properly when slip relations other than no-slip is used, Newton's method is applied. The derivative of the specific sub-field enthalpy with respect to the specific mixture enthalpy is then needed as we still solve for the change in specific mixture enthalpy, and is calculated as follows:

$$\frac{\partial h_f}{\partial h_k} = \frac{\partial h_f}{\partial T}\frac{\partial T}{\partial h_k} = \frac{\frac{\partial h_f}{\partial T}}{\frac{\partial h_k}{\partial T}} = \frac{\frac{\partial h_f}{\partial T}}{\sum_{f} \frac{M_f}{M_k}\frac{\partial h_f}{\partial T}} \tag{107}$$

The enthalpy temperature derivatives in equation (108) are calculated from the tabulated data in the specified PVT-files, or as the specified heat capacities for ideal fluids.

The fluid energy equations and the pipe wall energy equations are solved separately and repeatedly (due to the fluid-wall heat transfer), until the maximum residual is lower than a user-specified tolerance (default being 0.01).

The discretized version of the energy equation becomes (including the summation of the convective fluxes):

$$\begin{aligned}
&\frac{\left(M_{k,j}^{n+1}h_{k,j}^{n+1}+\frac{M_{k,j}^{n+1}}{2}\left(u_{k,j}^{n+1}\right)^{2}\right)-\left(M_{k,j}^{n}h_{k,j}^{n}+\frac{M_{k,j}^{n}}{2}\left(u_{k,j}^{n}\right)^{2}\right)}{\Delta t}\\
&+\sum_{f}\hat{m}_{f,J}^{n+1}A_{J}^{n}\left(u_{f,J}^{n+1}-u_{b,J}^{n+1}\right)\left(\hat{h}_{f,J}^{n+1}+\frac{1}{2}\left(u_{f,J}^{n+1}\right)^{2}\right)\\
&-\sum_{f}\hat{m}_{f,J-1}^{n+1}A_{J-1}^{n}\left(u_{f,J-1}^{n+1}-u_{b,J-1}^{n+1}\right)\left(\hat{h}_{f,J-1}^{n+1}+\frac{1}{2}\left(u_{f,J-1}^{n+1}\right)^{2}\right)=\\
&V_{k,j}^{n}\left(\frac{p_{j}^{n+1}-p_{j}^{n}}{\Delta t}-u_{b,j}^{n+1}\frac{p_{J}^{n+1}-p_{J-1}^{n+1}}{L_{j}^{n}}\right)\\
&-0.125W_{j,s}^{n}\rho_{g,j}^{n}\sum_{s}\left|u_{k,j}^{n}-u_{m,j}^{n}\right|\left(u_{k,j}^{n+1}-u_{m,j}^{n+1}\right)u_{l,j}^{n+1}\\
&+\sum_{m}Q_{k,m}+\dot{M}_{k,j}^{src}\left(h_{k,j}^{src}+\frac{\left(u_{k,j}^{src}\right)^{2}}{2}\right)-M_{k,j}^{n}u_{k,j}^{n+1}g\frac{1}{L_{j}}\int_{L_{j}}\sin\left(\theta\left(x\right)\right)dx\\
&\pm\phi_{e,j}^{n}\left(h_{d,j}^{n}+\frac{\left(u_{d,j}^{n}\right)^{2}}{2}\right)\mp\phi_{d,j}^{n}\left(h_{e,j}^{n}+\frac{\left(u_{e,j}^{n}\right)^{2}}{2}\right)
\end{aligned}\qquad(108)$$

The BDF2 and Crank-Nicolson time integration methods can also be used for the energy equation, see equation (29).
The integral that involves the inclination in the gravity term is not shown in discretized form, as it depends on the type of pipe being used (straight/bent).
The section control volume velocity $u_{b,j}^{n+1}$ in equation (109) is calculated as the average of the section borders.

The different heat transfer terms $Q_{k,m}$ are described in section 2.9.5.1.

There is no coupling between the wall layers in the axial direction, and hence the energy equation for each wall section can be solved independently.

Since we solve one energy equation for each mixture field and not for the individual sub-fields, we only know the mixture enthalpies. The specific mixture enthalpy itself can however not be used to calculate the temperature directly. It is however possible to calculate the individual specific sub-field enthalpies from the mixture, by assuming that all sub-fields in the mixture field have the same temperature. For instance for oil in mixture gas, water in mixture gas and continuous gas in mixture gas we then have:

$$T_{og} = T_{wg} = T_{gg} = T_g \tag{109}$$

Using the definition of the specific mixture enthalpy (equation (107)), we get an equation for the unknown temperature:

$$f\left(T_g^{n+1}\right) = M_{k,j}^{n+1} h_{k,j}^{n+1}\left(T_g^{n+1}\right) - \sum_f M_{f,j}^{n+1} h_{f,j}^{n+1}\left(T_g^{n+1}\right) = 0 \tag{110}$$

Here the masses and specific mixture enthalpy is known; only the individual specific sub-field enthalpies are unknown. This equation is easily solved with for instance Newton's method.

2.9.5.1 Heat transfer

The heat transfer terms in the energy equation require some special treatment in terms of implementation. One of the main difficulties regarding heat transfer lies in the fact that the sections in the pipeline might move during each time step (see section 2.11), while the wall sections for the pipeline walls remains fixed. There might then be a mismatch between these two grids, and the control volumes will generally have a varying number of neighbours that their energy equation is coupled to.

To ease the calculations of the heat transfer coefficients, the following class hierarchy is used:

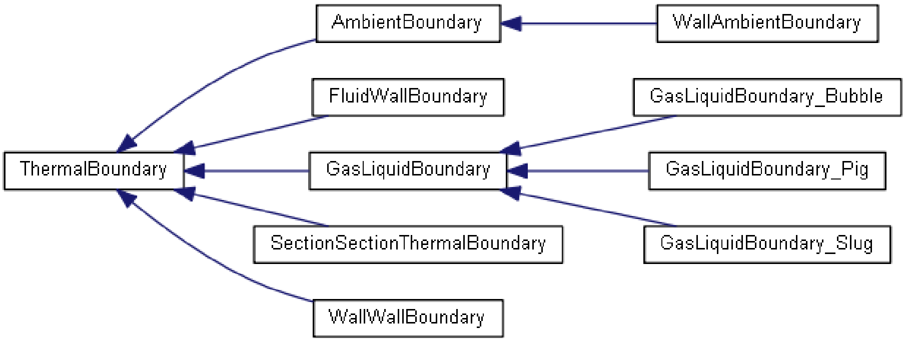


Figure 41: Class hierarchy for the thermal boundaries.

The base class *ThermalBoundary* represents a general heat transfer boundary between two control volumes, which may be either the gas or liquid within a section, or a pipe wall layer within a pipe wall section. The derived subclasses represent specific types of thermal boundaries (the class names are assumed to be self-explanatory). Before the energy equations can be solved, each section must perform some management related to *FluidWallBoundary* objects (boundaries between the fluids and the innermost pipeline wall). This management makes sure that each overlap (full or partial) between a section and the neighbouring pipe wall section is always represented by two *ThermalBoundary* objects, one for each fluid. This is illustrated in Figure 42.

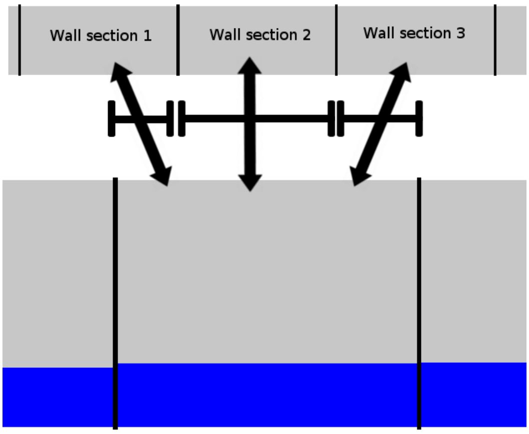


Figure 42: Schematic illustration of fluid-wall thermal boundaries. This section is in contact with three wall sections (two of them only partially), yielding three thermal boundaries between each mixture field and the wall.

The temperature is solved implicitly between the wall layers within one wall section, yielding an equation system with 1 lower and 1 upper diagonal, and as many equations are there are wall layers.

$$\begin{pmatrix} c_{1,\text{wall},1} & c_{1,\text{wall},2} & 0 & 0 \\ c_{2,\text{wall},1} & c_{2,\text{wall},2} & 0 & 0 \\ 0 & c_{3,\text{wall},2} & c_{3,\text{wall},3} & 0 \\ 0 & 0 & c_{4,\text{wall},3} & c_{4,\text{wall},4} \end{pmatrix} \begin{pmatrix} \Delta h_{wall,1} \\ \Delta h_{wall,2} \\ \Delta h_{wall,3} \\ \Delta h_{wall,4} \end{pmatrix} = \begin{pmatrix} \Delta rhs_{wall,1} \\ \Delta rhs_{wall,2} \\ \Delta rhs_{wall,3} \\ \Delta rhs_{wall,4} \end{pmatrix} \tag{111}$$

Heat transfer is calculated between different wall layers, between the innermost wall layer and the fluids, between the outermost wall layer and the ambient, between sections in the axial direction, and between the gas and liquid within one section. If no walls are specified, there will be no heat transfer with the surroundings.

The general formulation for steady state heat conduction between two objects can be expressed by the thermal resistivity and the temperature difference:

$$Q_1 = -Q_2 = \frac{T_1 - T_2}{R_{cond,tot}} \tag{112}$$

Here $R_{cond,tot}$ is the total combined thermal resistivity, and Q is the heat transfer (in Watt). T_1 is the temperature of the object to calculate the heat transfer for, and T_2 is the temperature of the other object involved in the heat exchange. Note that the heat gained by the first object is the same as the heat lost by the second object. The different calculations of the total thermal resistivity are shown in sections 2.9.5.1.1 - 2.9.5.1.3.

The heat transfer between the current object (*k*) and other object (*m*) is added to the right hand side and diagonal in the energy equation matrix as shown in equation (114):

$$\begin{aligned} h_{k,desired} &= h_k\left(p_k^{n+1}, T_m^{n+1}\right) \\ rhs_k &= \frac{h_{k,desired} - h_k^n}{R_{cond,km}^{n+1} c_k^{n+1}\left(p_k^{n+1}, T_k^{n+1}\right)} \\ diag_k &= \frac{1}{R_{cond,km}^{n+1} c_k^{n+1}\left(p_k^{n+1}, T_k^{n+1}\right)} \end{aligned} \tag{113}$$

The same procedure is also done for the other object (subscript *m*). In equation (114) $h_k\left(p,T\right)$ is a function that returns the specific mixture enthalpy (see equation (107)) for the mixture phase *k*, as function of pressure and temperature. The parameter c_k is the mixture heat capacity, defined by:

$$c_k^{n+1}\left(p,T\right) = \frac{1}{M_k^{n+1}} \sum_f M_f^{n+1} c_f\left(p^{n+1}, T^{n+1}\right) \tag{114}$$

The new value for the temperature and pressure is used as shown in equation (114) in each iteration, making the heat transfer implicit instead of explicit.

It is necessary to formulate the heat transfer coefficients as shown in equation (114) both to make the heat transfer to be correct when the mixture field consists of multiple fluids, and to make the calculations correct when using a real fluid with a PVT-file instead of an ideal fluid. When only one fluid is present in the mixture field and when using an ideal fluid (pipe walls and pigs will always be a single "fluid" with ideal fluid properties), equation (114) simplifies. The heat capacity is then constant, and the specific enthalpy can be calculated from:

$$h_{k,ideal}\left(T\right) = c_{k,ideal}T \tag{115}$$

We then get:

$$\begin{aligned} h_{k,desired} &= c_{k,ideal}T_m^{n+1} \\ rhs_k &= \frac{h_{k,desired} - h_k^n}{R_{cond,km}^{n+1}c_{k,ideal}} = \frac{T_m^{n+1}c_{k,ideal} - T_k^n c_{k,ideal}}{R_{cond,km}^{n+1}c_{k,ideal}} = \frac{T_m^{n+1} - T_k^n}{R_{cond,km}^{n+1}} \\ diag_k &= \frac{1}{R_{cond,km}^{n+1}c_{k,ideal}} \end{aligned} \tag{116}$$

This would give a change in specific enthalpy of:

$$\Delta h_k = \frac{rhs_k}{diag_k} = \frac{\dfrac{T_m^{n+1} - T_k^n}{R_{cond,km}^{n+1}}}{\dfrac{1}{R_{cond,km}^{n+1}c_{k,ideal}}} = c_{k,ideal}\left(T_m^{n+1} - T_k^n\right) \tag{117}$$

The thermal resistivity is not used in equation (118), but will be used if the diagonal and right hand side in the matrix contains values from other parts of the energy equation (which will always be the case). The thermal resistance then acts as a weight for the different energy exchange processes that are added to the matrix. Note also that the new value for the temperature of the other object is used. This is mainly important when the time step is large, and will then stabilize the heat transfer process. Testing showed that using the old temperature (explicit heat transfer) would give the correct trend for the temperature calculations, but with oscillations when the time step becomes large. Calculating the heat capacity for equation (114) based on new or old values for the temperature and pressure however showed little differences in the results (the mass should however be new values).

Note that pigs are treated similarly as the mixture liquid in a slug, but with the thermal conductance, heat capacity and surface conductivity as specified for the pig material.

2.9.5.1.1 Wall-wall heat transfer

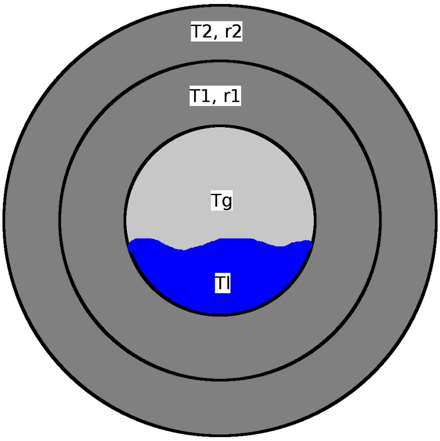


Figure 43: Illustration of a pipe with two pipe walls with different temperatures.

The heat equation at steady state for radial heat transfer in a cylinder [42] is given by equation (119):

$$\frac{1}{r}\frac{d}{dr}\left(kr\frac{dT}{dr}\right)=0 \qquad (118)$$

Here *k* is the thermal conductivity of the wall.

The radial temperature distribution can be found by integrating equation (119) twice:

$$T(r) = C_1 \ln r + C_2 \tag{119}$$

Using the boundary conditions $T(r_1) = T_1$ and $T(r_2) = T_2$ (see Figure 43) we get:

$$T(r) = \frac{T_1 - T_1}{\ln\left(\frac{r_1}{r_2}\right)} \ln\left(\frac{r}{r_2}\right) + T_2 \tag{120}$$

Fourier's law gives the heat transfer as function of the temperature distribution:

$$Q_r = q_r A = -kA\frac{dT}{dr} \tag{121}$$

Here $A = 2\pi rL$ is the surface area for a wall segment of length L.
Combining Fourier's law and equation (121), we can find the heat flux between two wall layers from:

$$\begin{aligned} Q_r &= -\frac{T_1 - T_2}{R_{cond,tot}} \\ \frac{1}{R_{cond,tot}} &= \frac{\ln\left(\frac{r_{12}}{r_1}\right)}{k_1 2\pi L} + \frac{\ln\left(\frac{r_2}{r_{12}}\right)}{k_2 2\pi L} + \frac{1}{h_{contact}} \end{aligned} \tag{122}$$

Here r_1 is at the middle of the inner wall layer, r_{12} is at the connection between the wall layers, and r_2 is at the middle of the outer wall layer. $h_{contact}$ is the thermal contact conductance, which can optionally be specified for the walls. The contact between two walls will typically never be a perfect smooth surface, but consist of gaps and contact points. This value typically has to be determined experimentally.

Currently there is no heat transfer between walls in the axial direction, but this will typically be much smaller than the radial heat transfer. The ratio of the areas where the radial and axial heat transfer occur is of order of wall length to wall thickness L/w, which is typically several orders of magnitude.

2.9.5.1.2 Fluid-wall heat transfer

For the heat transfer between the mixture gas and the pipe wall the properties of the continuous gas field is used, while for the mixture liquid the contributions from the continuous oil and continuous water fields are summed up.
The heat transfer between the innermost wall layer and field *f* is calculated as follows:

$$\frac{1}{R_{cond,tot,f}} = \frac{\frac{1}{\pi D L h_{f,wall}} + \frac{\ln\left(\frac{D + w_{wall}}{D}\right)}{2\pi L k_{wall}}}{\frac{S_f}{\pi D}} \tag{123}$$

Here k_{wall} and w_{wall} are the thermal conductivity and width/thickness of the innermost wall layer respectively. The factor $\frac{S_f}{\pi D}$ is the ratio of the surface area covered by the fluid to the total pipe surface area, a factor that should be multiplied by each of the area in each individual thermal resistivity. For single phase flow this factor will be one, while it is zero when the phase fraction goes to zero. To account for bent pipes and section spanning multiple pipes, the perimeters used to calculate the surface area are integrated/summed over the different sub-control volumes, taking into account the weighting between annular and stratified flow as function of inclination.
The parameter $h_{k,wall}$ is the surface conductivity between field *f* and the wall, which can be specified by the user.

If not specified, the surface conductivity is calculated from the correlation of Gnielinski [43] if the Reynolds number is above 2300, and linear interpolation between the laminar value [42] and the Gnielinski correlation if the Reynolds number is below 2300:

$$
\begin{aligned}
&\mathrm{Nu}_{forced,lam} = 3.66 \\
&\mathrm{Nu}_{forced,tur}\left(\mathrm{Re}_f\right) = \frac{0.125\lambda_f \mathrm{Re}_f \mathrm{Pr}_f}{1+12.7\left(0.125\lambda_f\right)^{\frac{1}{2}}\left(\mathrm{Pr}_f^{\frac{2}{3}}-1\right)} \\
&\mathrm{Nu}_{forced} = \begin{cases} \mathrm{Nu}_{tur} & \mathrm{Re}_f > 2300 \\ \dfrac{\mathrm{Re}_f}{2300}\left(\mathrm{Nu}_{tur}\left(2300\right)-\mathrm{Nu}_{lam}\right)+\mathrm{Nu}_{lam} & \mathrm{Re}_f \le 2300 \end{cases} \\
&h_{f,wall,forced} = \frac{k_f \mathrm{Nu}}{D_{h,f}}
\end{aligned}
\tag{124}
$$

Here, Nu is the Nusselt number, the ratio of convective to conductive heat transfer, while λ_f is the friction factor calculated using the Haaland correlation (equation (145)). D_h is the hydraulic diameter, and Pr is the Prandtl number, defined by the ratio of momentum diffusivity (kinematic viscosity) to thermal diffusivity:

$$
\mathrm{Pr}_f = \frac{c_f \mu_f}{k_f} \tag{125}
$$

All variables in equation (125) are evaluated at time step *n*+1, and the average temperature between the fluid and the wall is used in all calculations where the temperature is needed. Note that this surface conductance value is for *forced* convection.

For *free*/*natural* convection, a correlation found in the LedaFlow user manual is used [44], based on correlations for a vertical plate with height H adapted to a circular pipe:

$$
\begin{aligned}
&H = \min\left(5D, \frac{D}{\cos(\theta)}\right) \\
&\mathrm{Gr} = \frac{g\rho_f H^3 \left|T_{wall} - T_f\right| \left|\frac{\partial \rho_f}{\partial T}\right|}{\mu_f^2} \\
&\mathrm{Nu}_{free,lam} = \frac{3}{4}\left(\frac{2\,\mathrm{Pr}_f}{5\left(1+2\,\mathrm{Pr}_f^{0.5}+2\,\mathrm{Pr}_f\right)}\right)^{0.25} \left(\mathrm{Gr}_f\ \mathrm{Pr}_f\right)^{0.25} \\
&\mathrm{Nu}_{free,tur} = \left(\frac{\mathrm{Gr}_f\ \mathrm{Pr}_f}{300\left(1+\left(\frac{0.5}{\mathrm{Pr}_f}\right)^{\frac{9}{16}}\right)^{\frac{16}{9}}}\right)^{\frac{1}{3}} \\
&\mathrm{Nu}_{free} = \mathrm{Nu}_{free,lam} + \frac{\mathrm{Nu}_{free,tur} - \mathrm{Nu}_{free,lam}}{1+\exp\left(1-\frac{\mathrm{Gr}_f\ \mathrm{Pr}_f}{2\cdot 10^{10}}\right)} \\
&h_{f,wall,free} = \frac{k_f \mathrm{Nu}_{free}}{H}
\end{aligned}
\qquad (126)
$$

Here Gr is the Grashof number, the ratio of buoyancy to viscous forces.

Finally, the combined surface conductance between the fluid and wall is calculated from:

$$
h_{f,wall} = \sqrt{h_{f,wall,forced}^2 + h_{f,wall,free}^2} \qquad (127)
$$

2.9.5.1.3 Wall-ambient heat transfer

The heat transfer between the outermost wall layer and the ambient is calculated similarly as the heat transfer between the fluids and the innermost wall layer:

$$\frac{1}{R_{cond,tot}} = \frac{1}{\pi D_{wall,outer} L h_{wall,amb}} + \frac{\ln\left(\frac{D_{wall,outer}}{D}\right)}{2\pi L k_{wall}} \tag{128}$$

Here $D_{wall,outer}$ is the diameter of the outermost wall layer, while $h_{wall,amb}$ is the user-specified surface conductivity between the wall and the ambient fluid. If specified to not use user-specified U-values, the wall-ambient heat transfer coefficient is calculated from the correlation of Churchill and Bernstein [45]

$$\begin{aligned}
\mathrm{Nu}_{wall,amb} &= 0.3 + \frac{0.62\,\mathrm{Re}_{amb}^{\frac{1}{2}}\,\mathrm{Pr}_{amb}^{\frac{1}{3}}}{\left(1+\left(\frac{0.4}{\mathrm{Pr}_{amb}}\right)^{\frac{2}{3}}\right)^{\frac{1}{4}}}\left(1+\left(\frac{\mathrm{Re}_{amb}}{282000}\right)^{\frac{5}{8}}\right)^{\frac{4}{5}} \\
\mathrm{Re}_{amb} &= \frac{\rho_{amb} u_{amb} D_{wall,outer}}{\mu_{amb}} \\
h_{wall,amb} &= \frac{\mathrm{Nu}_{wall,amb} k_{amb}}{D_{wall,outer}}
\end{aligned} \tag{129}$$

Note that equation (130) needs certain variables describing the ambient fluid, in addition to the ambient temperature: The density, viscosity, thermal conductivity, heat capacity, and velocity of the ambient fluid.

Since these variables are given constant values, the result from equation (130) will be the same constant through the entire simulation.

2.9.5.1.4 Radial fluid-fluid heat transfer

For the heat transfer between the mixture gas and mixture liquid the properties of the continuous gas field and continuous oil field is used if oil is present. If oil is not present, the continuous water field is used for the liquid.
The gas-liquid thermal resistivity is calculated by:

$$R_{cond,tot} = \max\left(\frac{1}{h_{gl}S_{gl}L}, \frac{1}{h_{gl}S_{gl,\min}L}\right) \tag{130}$$

Here h_{gl} is the user-specified gas-liquid thermal conductivity S_{gl} is the gas-liquid interfacial perimeter and L is the length of the section. $S_{gl,\min}$ is a minimum gas-liquid interfacial perimeter calculated from a holdup value of 0.01, used to set a lower limit on the heat transfer. This will make a mixture field with a low phase fraction obtain the temperature of the other mixture field.

2.9.5.1.5 Axial fluid-fluid heat transfer

Heat transfer is also calculated between continuous fields of the same type in the axial direction.
The thermal resistivity for field *k* between section *j* and the next section (*j*+1) is calculated using the average of the neighbouring phase areas as the contact area:

$$\frac{1}{R_{cond,tot,k}} = \sum_{f}\left[\frac{1}{0.5\left(A_{f,j} + A_{f,j+1}\right)}\left(\frac{0.5L_j}{k_{f,j}} + \frac{0.5L_{j+1}}{k_{f,j+1}}\right)\right] \tag{131}$$

This heat transfer will typically be much smaller than the radial heat transfer, but can be important for instance if a pipeline is shut down.

2.9.6 Boundary conditions

Two types of boundary conditions are available in the presented model: An open boundary with a constant pressure (typically the outlet), and a closed boundary with specified mass rates entering (typically the inlet).
The temperature is user-specified and is assumed constant in both the inlet and outlet section regardless of the type of boundary condition that is used.

The user can specify the length of the inlet and outlet sections directly. If not specified, the lengths are calculated as the average between the minimum and maximum bubble length coefficients. It is also possible to update the lengths of the inlet and outlet sections to that of the neighbouring sections during the simulation.

2.9.6.1 Constant pressure boundary

The constant pressure boundary needs a user-specified pressure. In the following examples the outlet is assumed to be the constant pressure boundary, but the inlet can be treated similarly. To obtain a smooth pressure, holdup and velocity profile at the open outlet boundary the part of the momentum control volume for the border at the pressure boundary has zero extent into the outlet section, which makes the velocity at the control volume end-point being chosen from the outlet border. This is equivalent to using a Neumann boundary condition with a zero derivative for the velocity, holdup and enthalpy, and is just a relatively simple way of obtaining this with the functionality of changing the control volume extent. The pressure gradient will not be zero, and the specified outlet pressure will be used as the pressure at the outlet border. Since the control volume end-point is at a border, the specific mass is unknown and must be determined using a flux limiter. If the flow is going out of the system the specific mass will be chosen from inside the pipe, and if the flow is going into the system the specific mass calculated from the user specified holdup values in the outlet section are used (default being an empty cell with only gas). For a description of the extent of the momentum control volumes see for instance Figure 6, and for an illustration of the smooth profiles obtained with this boundary condition for a horizontal test case see Figure 44 and

Figure 53. The test case is described in section 2.16.2. The small deviations from the linear fit seen at the outlet in Figure 44 most likely comes from the effect of assuming a zero derivative of the holdup at the outlet.

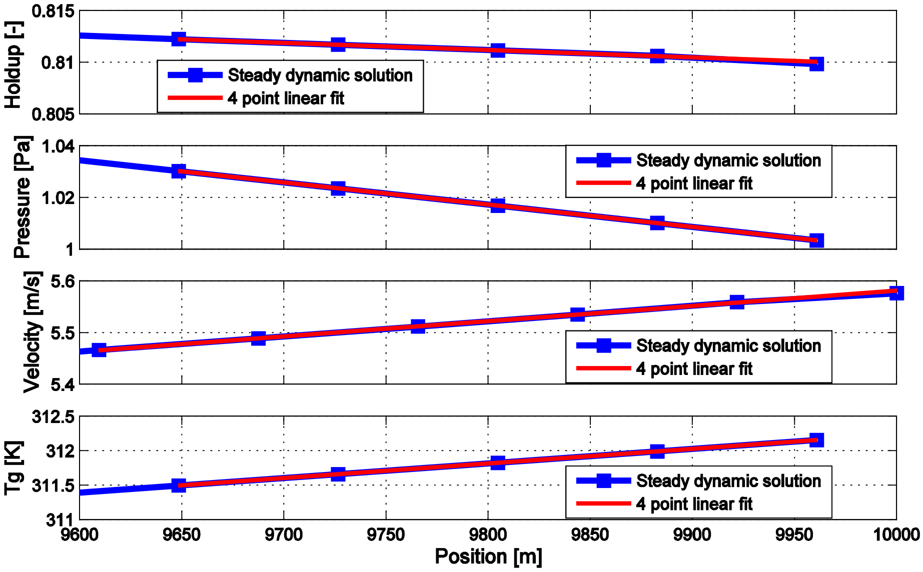


Figure 44: The end of the pipeline in a transient simulation of a horizontal pipe, at steady state.

2.9.6.2 Mass flow boundary

For a closed inlet or outlet border the gas, oil and water mass rates needs to be specified (default is 0). In the following examples it is assumed that the inlet is the closed mass flow boundary, but the outlet can be treated similarly. Since the variable we solve the momentum equation for is velocity and not momentum, we need the velocities at the mass flow boundary. These are calculated from the mass flow rate, phase area and density as follows:

$$u_k = \frac{\dot{M}_k}{A\alpha_k \rho_k} \tag{132}$$

The mixture liquid velocity is calculated as the mass weighted velocity from the continuous oil and water fields (only continuous field mass rates can be specified at the boundary). One important question is what density and holdup to use for the inlet section in equation (133), for which there are several possibilities. One possibility is to calculate the inlet holdup (and pressure) from the steady state solver. Note that this does not mean that the remaining sections in the pipeline will start from the steady state solution. Another possibility that is implemented is for the user to specify the inlet pressure directly, and calculate the holdups from the unit-cell model (this is the default option). There is also a possibility implemented where the mass (and holdup and pressure) in the inlet section are copied from the neighbouring section at the end of each time step (this is enabled as default as well). The phase fractions that are copied into the inlet section are limited not to be too small, to avoid unrealistically large velocities (since we divide by the holdup in equation (133)). Each phase fraction is limited to be minimum 0.5 times the no-slip phase fraction (ratio of phase volume flow to total volume flow). Note that this will only make the phase fraction and velocity change at the inlet, while the mass flow rate will remain constant. An option is also available where constant user-specified holdups are used in the inlet section. This functionality is needed for instance in the water-faucet test case presented in section 2.6.

It is also possible to add multiple mass sources or sinks at any location in the pipe.

2.9.6.3 Periodic boundaries

It is also possible to use periodic boundaries in the Sluggit code. The inlet and outlet border will then be the same border object. This leads to the first and last section in the pipeline affecting each other, and one will get coefficients in the corners of the matrix of equations. There are two options to deal with such a matrix with corner elements, as the matrix can no longer be stored as a banded matrix (at least with the banded matrix format being used in Sluggit):
Either the entire full matrix must be stored, or the procedure of Sherman-Morrison can be used, which is faster and requires less storage. The Sherman-Morrison method is described in section 2.9.7.1. In a periodic system there will then be no pressure boundaries, and no mass flow boundaries. To inject or remove fluids from the pipeline one would thus have to use

mass sources or sinks. One might also need a driving force, at least in a horizontal case for the fluids to move. The user specified force described in section Appendix C.4 can be used for this purpose.

Figure 45 shows a wave in a periodic system, taken from a test case from Paper 2.

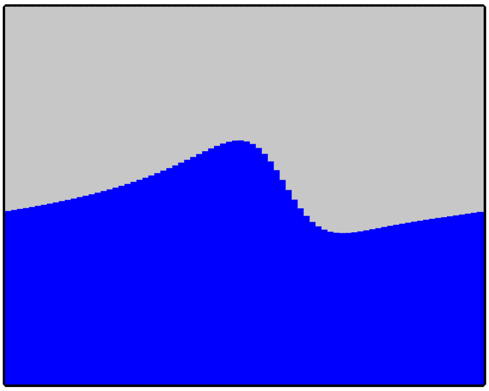


Figure 45: Image of a wave in a horizontal pipe with periodic boundaries and a driving pressure force.

2.9.7 Generic equation class

To ease the calculation of which index in the matrix system of equations to access when updating the different equations, a generic equation class was constructed (named *NPhaseEquation*). The inheritance diagram for this class is shown in Figure 46, showing all of the different equation types used in Sluggit.

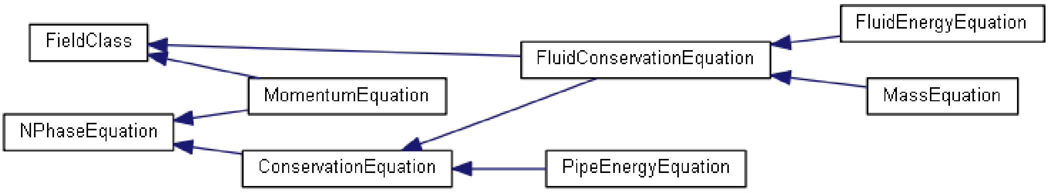


Figure 46: Inheritance diagram for the generic equation class.

Both the mass equation and fluid energy equation have their values stored in the sections, and derive from the fluid conservation equation, which keeps a pointer to the section they represent. Both the fluid conservation equation and the pipe energy equation inherit from the conservation equation class, which is used to update the energy equation systems. Both the conservation equation and momentum equation classes in turn inherit from the general equation class (*NPhaseEquation*) described in this section. In addition, the momentum equation and fluid conservation equation inherit from the field class, which is used to keep track of which phase the equation represents (mixture gas, mixture oil, mixture water, or any of the sub-fields). The pipe energy equation however does not inherit from the field class, as it does not represent any fluid.

The details of how the indexes for the coefficients in the banded matrix are calculated will be described here, as it might be of interest for others working on similar problems.
The generic equation class takes two inputs, the first being the number of variables to solve for, which will be referred to by $N_{variables}$. This number will be one for the mass equation, two for a two-phase momentum equation, three for a three-phase momentum equation, etc. Furthermore this class requires an integer input called the *variable index*, which will be referred to by the symbol $N_{k,index}$ for the variable that belongs to this equation, and $N_{m,index}$ for the other variables to be solved. The variable index parameter is mainly used to calculate the offset in the arrays between different variables. For a three-phase system one can for instance give the gas a variable index of 0, oil could be 1, and water 2. The numbers could just as well have been 10, 11 and 12, what is important is the offset between the numbers. In the current C++ code the variable indexes are simply a list of enumerated values describing the different sub-fields and mixture fields.
In addition, it is possible to specify how many neighbouring cells are affected by this equation ($N_{neighbor}$). The default is one (only nearest neighbours) for both the mass, momentum and energy equations, but for higher order convection schemes this number could be higher (typically two, meaning next nearest neighbours are included).

The number of diagonals in the banded matrix can be calculated from:

$$N_{diag} = N_{variables}\left(N_{neihgbor} + 1\right) - 1 \tag{133}$$

It is here assumed a symmetric banded matrix with the same number of lower and upper diagonals, which will be the case for the two-fluid model.

The number of columns required for storing the non-zero coefficients in the banded matrix can now be calculated from equation (135):

$$N_{column} = 2N_{diag} + 1 \tag{134}$$

For the mass equation and a two-phase momentum equation affecting only nearest neighbouring sections, this gives three and seven columns respectively.

The number of rows must however be calculated by custom functions for each equation type, and will depend on the number of control volumes for the different equations (which might change in the Sluggit code due to grid management). If the full matrix is stored instead of a banded matrix, the number of columns will equal the number of rows.

Before updating the different equations, each equation must be connected/appended as a new row in the system of equations. When doing this, information about which row this equation is located at will be stored in the generic class (n_{row}).

The different indexes in the current row to be accessed can now easily be calculated from the available information. Note that for some of the coefficients, the variable indexes must be specified. An example of this can be when accessing the coefficient for gas from the liquid momentum equation, for instance when updating the effect from interfacial friction.

Table 3: How to calculate the index in an equation row for different equation coefficients.

Coefficient	Index
Current variable, left equation	$N_{diag} - \left(N_{neihgbor} N_{variables}\right)$
Other variable, left equation	$N_{diag} - \left(N_{neihgbor} N_{variables}\right) + N_{k,index} - N_{m,index}$
Current variable, current equation	$N_{diag} = n_{row}$
Other variable, current equation	$N_{diag} + N_{k,index} - N_{m,index}$
Current variable, next equation	$N_{diag} + \left(N_{neihgbor} N_{variables}\right)$
Other variable, next equation	$N_{diag} + \left(N_{neihgbor} N_{variables}\right) + N_{k,index} - N_{m,index}$

If the full matrix is stored instead of a banded matrix, a banded matrix index ind_{band} can be converted to a full matrix index ind_{full} from:

$$ind_{full} = n_{row} + ind_{band} - N_{diag} \tag{135}$$

For periodic boundaries, the modulus operator must also be applied to the matrix index:

$$ind_{full} = \text{mod}\left(ind_{full}, N_{column}\right) \tag{136}$$

$$\text{mod}(x, y) = x - \left(\text{floor}(x / y)\right) y \tag{137}$$

An example of how the banded matrix for the change in gas and liquid velocity for the momentum equation looks is shown in equation (139). The number of variables to solve for is 2 and the number of affected neighbours is 1, so the number of upper and lower diagonals according to equation (134) will be $2(1+1)-1=3$. The number of columns in the banded matrix to store the non-zero coefficients becomes according to equation (135): $2 \cdot 3 + 1 = 7$.

Note that the coefficients marked with * will only be non-zero for a system with periodic boundaries.

$$\begin{pmatrix}
c_{l,1,1} & c_{g,1,1} & c_{l,1,2} & c_{g,1,2} & & & & & c^{*}_{l,1,N} & c^{*}_{g,1,N} \\
c_{l,2,1} & c_{g,2,1} & c_{l,2,2} & c_{g,2,2} & & & & & c^{*}_{l,2,N} & c^{*}_{g,2,N} \\
c_{l,3,1} & c_{g,3,1} & c_{l,3,2} & c_{g,3,2} & c_{l,3,3} & c_{g,3,3} & & & & \\
c_{l,4,1} & c_{g,4,1} & c_{l,4,2} & c_{g,4,2} & c_{l,4,3} & c_{g,4,3} & & & & \\
 & & c_{l,5,2} & c_{g,5,2} & c_{l,5,3} & c_{g,5,3} & c_{l,5,4} & c_{g,5,4} & & \\
 & & c_{l,6,2} & c_{g,6,2} & c_{l,6,3} & c_{g,6,3} & c_{l,6,4} & c_{g,6,4} & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & & & c_{l,N-3,N-2} & c_{g,N-3,N-2} & c_{l,N-3,N-1} & c_{g,N-3,N-1} & c_{l,N-3,N} & c_{g,N-3,N} \\
 & & & & c_{l,N-2,N-2} & c_{g,N-2,N-2} & c_{l,N-2,N-1} & c_{g,N-2,N-1} & c_{l,N-2,N} & c_{g,N-2,N} \\
c^{*}_{l,N-1,1} & c^{*}_{g,N-1,2} & & & & & c_{l,N-1,N-1} & c_{g,N-1,N-1} & c_{l,N-1,N} & c_{g,N-1,N} \\
c^{*}_{l,N,1} & c^{*}_{g,N,2} & & & & & c_{l,N,N-1} & c_{g,N,N-1} & c_{l,N,N} & c_{g,N,N}
\end{pmatrix}
\begin{pmatrix}
\Delta u_{l,1} \\ \Delta u_{g,1} \\ \Delta u_{l,2} \\ \Delta u_{g,2} \\ \vdots \\ \Delta u_{l\,N-1} \\ \Delta u_{g\,N-1} \\ \Delta u_{l\,N} \\ \Delta u_{g\,N}
\end{pmatrix}
= \begin{pmatrix}
rhs_{l,1} \\ rhs_{g,1} \\ rhs_{l,2} \\ rhs_{g,2} \\ \vdots \\ rhs_{l,N-1} \\ rhs_{g,N-1} \\ rhs_{l,N} \\ rhs_{g,N}
\end{pmatrix} \quad (138)$$

2.9.7.1 Matrix solvers

The system of equations is solved by LU-decomposition for both banded and full matrixes. According to [46], the number of operations needed to solve a banded system of equations with LU-decomposition is $(3N-2)(N_{variables}^{3}+N_{variables}^{2})$, where N is the number of rows. For a full matrix on the other hand, the number of operations is of order $\frac{2}{3}N^{3}$. When solving a system with periodic boundary conditions, corner elements appears as illustrated in equation (139). Solving such a matrix would typically result in filling in non-zero values in large parts of the matrix, and require similar number of operations as a full matrix. Instead of doing this, the Sherman-Morrison algorithm has been implemented to handle periodic systems [47]. This method looks at the matrix to be solved ($\mathbf{A}$) as a banded matrix, plus the product of two vectors containing the corner elements:

$$\mathbf{A}=\mathbf{A}^{*}+\mathbf{u}\mathbf{v}^{\mathrm{T}} \tag{139}$$

$$\mathbf{u}=\begin{bmatrix}\gamma\\0\\.\\.\\A_{N-1,1}\\A_{N,1}\end{bmatrix}\quad\mathbf{v}=\begin{bmatrix}1\\0\\.\\.\\A_{1,N-1}/\gamma\\A_{1,N}/\gamma\end{bmatrix} \tag{140}$$

The matrix $\mathbf{A}^{*}$ contains the banded part of $\mathbf{A}$, slightly modified: The coefficients at (1,1) and at $\left(N-N_{variables},N-N_{variables}\right)$ to $\left(N,N\right)$ are modified. The coefficient γ can be arbitrarily chosen as it only affects the numerical accuracy, and is currently set to 1.

The solution to $\mathbf{Ax} = \mathbf{b}$ can then be found by first solving $\mathbf{A}^*\mathbf{y} = \mathbf{b}$ and $\mathbf{A}^*\mathbf{z} = \mathbf{u}$, and then computing the solution $\mathbf{x}$ as:

$$\mathbf{x} = \mathbf{y} - \left[\frac{\mathbf{v}^{\mathrm{T}} \cdot \mathbf{y}}{1 + \mathbf{v}^{\mathrm{T}} \cdot \mathbf{z}}\right]\mathbf{z} \tag{141}$$

Note that the two dot-products in equation (142) forms two sums that must be calculated first. The method only works to remove one row of the lower left corner elements and one column of the upper right corner elements, but can be called recursively to remove larger blocks of corner elements. Equation (141) shows the $\mathbf{u}$ and $\mathbf{v}$ vectors after removing the lower left row and upper right column of the corner elements. In general, the coefficients γ and 1 in the $\mathbf{u}$ and $\mathbf{v}$ vectors will be at the same position as the row/column of corner elements to be removed. Above this index $\mathbf{u}$ and $\mathbf{v}$ are zero, while the bottom coefficients equals the lower left and upper right parts of $\mathbf{A}$ to be removed. $\mathbf{A}^*$ will be similar to $\mathbf{A}$, but with γ subtracted at the same diagonal index as were it is located in $\mathbf{u}$. In addition, the vector product of $\mathbf{uv}^{\mathrm{T}}$ is subtracted in the bottom right corner. For the matrix shown in equation (139), calling the Sherman-Morrison method once will create a matrix $\mathbf{A}^*$ with $c^*_{l,N-1,1}$, $c^*_{l,N,1}$, $c^*_{l,1,N}$ and $c^*_{g,1,N}$ removed. When then trying to solve $\mathbf{A}^*\mathbf{y} = \mathbf{b}$ and $\mathbf{A}^*\mathbf{z} = \mathbf{u}$, $\mathbf{A}^*$ still contains some corner elements, and the Sherman-Morrison method is applied once more. For a system of equations where $N_{\mathrm{var}\mathit{iables}}$ is the number of variables to solve for, the blocks of corner elements will be of size $N_{\mathrm{var}\mathit{iables}} \times N_{\mathrm{var}\mathit{iables}}$. This requires a total of $2^{N_{\mathrm{var}\mathit{iables}}} - 1$ calls to the Sherman-Morrison routine, resulting in approximately $\left(2^{N_{\mathrm{var}\mathit{iables}}} - 1\right)(3N-2)(N_{\mathrm{var}\mathit{iables}}{}^3 + N_{\mathrm{var}\mathit{iables}}{}^2)$ operations. This is typically far less than the $\frac{2}{3}N^3$ operations if using the full matrix.

Figure 47 shows a comparison of the performance of the Sherman-Morrison method and the LU-decomposition method using a full matrix for some test matrixes, and the theoretical values. The obtained simulation time ratios does not match exactly the theoretical ones, most likely because the ratio from the simulations are based on the computational time while the theoretical ratios are calculated from the number of operations. The results however seem to follow the correct trend, showing that it is possible to obtain a gain of several orders of magnitude for large periodic systems.

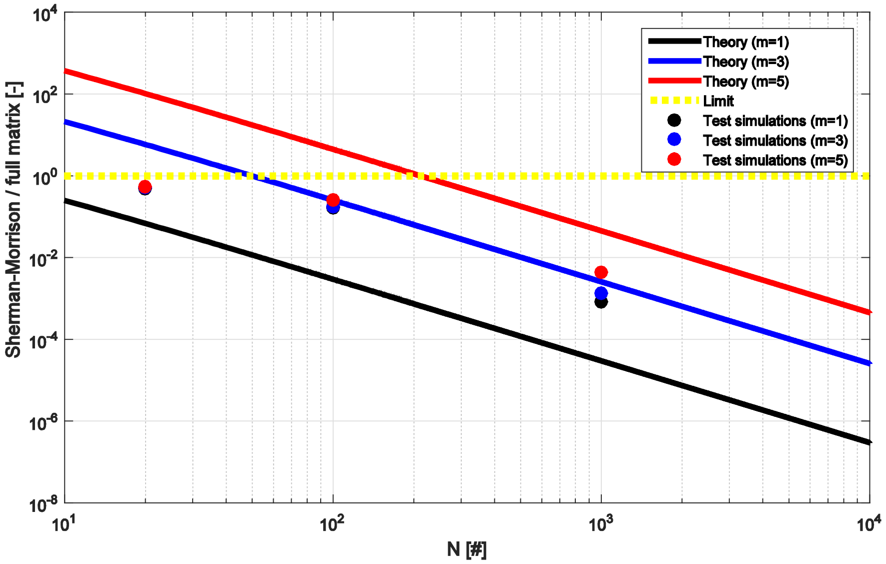


Figure 47: Theoretical and simulated values for the relative computational time between the Sherman-Morrison method and LU-decomposition using the full matrix.

2.10 Closure models

This section presents the necessary closure models required to close the system of equations, like friction and entrainment models.

2.10.1 Friction models

Friction models are needed to account for the friction between the gas and liquid phases and the pipe wall, and for the friction at the interphase between the gas and liquid.

The wall shear stress is expressed by the Darcy friction factor definition [48]:

$$\tau_{k,wall} = \frac{1}{8}\lambda_{k,wall}\rho_k u_k \left|u_k\right| \tag{142}$$

Here $\lambda_{k,wall}$ is the friction factor, which is described in more detail by equations (145) and (146). The interfacial shear stress is modelled similarly to the wall friction:

$$\tau_i = \frac{1}{8}\lambda_i \rho_g \left(u_k - u_m\right)\left|u_k - u_m\right| \tag{143}$$

For friction factors, we use the Haaland and Hagen-Poiseuille friction factors ([41, 49]), for turbulent and laminar flow respectively:

$$\lambda_{k,wall} = \left(-1.8\log_{10}\left[\frac{6.9}{\mathrm{Re}_k} + \left(\frac{\varepsilon / D_{h,k}}{3.7}\right)^{1.11}\right]\right)^{-2} \tag{144}$$

$$\lambda_{k,wall} = \frac{64}{\mathrm{Re}_k} \tag{145}$$

Here ε is the hydraulic roughness, and Re_k is the Reynolds number for phase *k*:

$$\mathrm{Re}_k = \frac{\rho_k D_{h,k} u_k}{\mu_k} \tag{146}$$

The parameter $D_{h,k}$ is the hydraulic diameter for phase *k*, while μ_k represents the dynamic fluid viscosity. For the gas phase, we use:

$$D_{h,k} = \frac{\pi D^2}{S_g + S_i} \tag{147}$$

For the liquid, we use:

$$D_{h,k} = \frac{\pi D^2}{S_l} \tag{148}$$

To achieve continuity, we use the maximum value of the laminar and turbulent friction factors, as long as the Reynolds number is above a minimum limit. The minimum limit is set to 300 in the Sluggit code. For stratified flow we apply the interfacial friction factor correlation of Andritsos and Hanratty [50]:

$$\begin{array}{ll} \lambda_i = \lambda_{g,0} & \alpha_g u_g < u_{g,crit} \\ \lambda_i = \lambda_{g,0}\left[1 + 15\sqrt{\frac{h_l}{D}}\left(\frac{\alpha_g u_g}{u_{g,crit}} - 1\right)\right] & \alpha_g u_g \geq u_{g,crit} \end{array} \tag{149}$$

The critical velocity $u_{g,crit}$ is defined as:

$$u_{g,crit} = 5\sqrt{\frac{\rho_{g,atm}}{\rho_g}} \tag{150}$$

Here $\rho_{g,atm}$ is the gas density at atmospheric pressure.

For annular flow, we use the correlation of Moeck [51]:

$$\lambda_i = 0.02\left(1 + 1458\left(\frac{\delta_l}{D}\right)^{1.42}\right) \tag{151}$$

Here, δ_l is the mean film thickness in annular flow (see equation (40)).

Furthermore, we limit the relative film thickness $\frac{\delta_l}{D}$ in equation (152) to be maximum 0.2, which is the maximum experimental value the correlation is defined for.

The interfacial friction factor is weighted between the stratified and annular models by using the inclination (see section 2.9.4.1).

2.10.1.1 Pig friction

The wall friction for pigs is modelled after the relatively simple model of Cordell [52], where the driving force of the pig equals the friction force:

$$\begin{aligned} \Delta P \frac{\pi D^2}{4} &= FRICTION \cdot \pi D \\ \Delta P &= \frac{K}{D} \end{aligned} \tag{152}$$

This gives a constant friction force per circumference. The factor *K* is most likely a function of amongst others the pig material, and will be different for different pigs. The pig friction is consequently given by:

$$F_{fric} = \left(\frac{K}{D}\right)\frac{V_{pig}}{L_{pig}} \tag{153}$$

The wall friction is thus independent of velocity, and will be applied as an explicit term. The sign of the friction term is however taken from the current velocity, or from the current pressure gradient over the pig if the current velocity is zero:

$$F_{wall,pig}^{n+1} = \begin{cases} \text{sign}\left(-\Delta p_{pig}^{n}\right) F_{fric} & u_{pig}^{n} == 0 \\ \text{sign}\left(u_{pig}^{n}\right) F_{fric} & \text{otherwise} \end{cases} \tag{154}$$

The wall friction for the pigs will be applied after all other updates has been done to the momentum equation. One can then use the current value of the right hand side if the momentum equation as an estimate of the other forces acting on the pig. If the wall friction force is larger than the other forces (smaller than if it the velocity negative), the equation coefficients are reset and the pig is explicitly given a velocity of zero (as described in section 2.9.4.2). This will typically happen when then simulation starts: The pressure force will build up until it overcomes the friction forces on the pig, and first then will the pig start to move.

The mass and velocity of the pig take the same place as the mixture liquid velocity and oil mass in the system of equations. The mass of a solid pig should not change during a simulation, so the pig borders will travel with the mixture liquid velocity, giving a zero mass flux at the borders.

2.10.2 Entrainment and deposition rates

It is possible to include multiple dispersed/entrained fields in the Sluggit model: Both liquid droplets in the gas, and gas in liquid in both slugs and in the bubble sections. Note that all types of entrainment and deposition described in this section are handled as source terms in the mass, momentum, pressure and energy equations.

2.10.2.1 Liquid droplets

For entrainment in both horizontal and vertical flow and for deposition in vertical flow we use the correlations of Govan [53], presented for entrainment and deposition of oil in equation (156). The entrainment correlation for vertical flow has been multiplied by a factor of 2 compared to the original correlation presented by Govan, as this was found to give better predictions of the slugging periods and liquid holdup in testing against experimental data. For deposition in horizontal flow a linear relationship with the deposition constant and droplet field concentration is used. Like for the wall and interfacial friction, the entrainment rates are weighted between the horizontal and vertical models by the weighting factor in equation (78) as an attempt to interpolate the entrainment and deposition rates for inclinations between horizontal and vertical configurations.

$$
\begin{aligned}
&\phi_{e,v} = LS \cdot 2 \cdot 5.75 \cdot 10^{-5} \left[\left(\max\left[\mathrm{Re}_{ol} - \mathrm{Re}_{crit}, 0 \right] \right)^2 \frac{\mu_{ol}^2 \rho_{ol}}{D \sigma_{go} \rho_{gg}^2} \right]^{0.316} \alpha_{gg} \rho_{gg} \left| u_{gg} \right| \\
&\mathrm{Re}_{ol} = \frac{\alpha_{ol} \rho_{ol} \left| u_{ol} \right| D}{\mu_{ol}} \quad \mathrm{Re}_{crit} = \exp^{5.8504 + 0.4249 \frac{\mu_{gg}}{\mu_{ol}} \sqrt{\frac{\rho_{ol}}{\rho_{gg}}}} \\
&\phi_{d,v} = \begin{cases} LS \cdot 0.18 C_{og}, & \dfrac{C_{og}}{\rho_{gg}} < 0.3 \\ LS \cdot 0.083 \left(\dfrac{C_{og}}{\rho_{gg}} \right)^{-0.65} C_{og}, & \dfrac{C_{og}}{\rho_{gg}} > 0.3 \end{cases} \\
&\phi_{e,h} = 0.5 \phi_{e,v} \\
&\phi_{d,h} = LS \cdot k_{D,h} C_{og}
\end{aligned}
\tag{155}
$$

The entrainment and deposition rates for water are calculated similarly as for oil. In equation (156) σ_{go} is the gas-oil interfacial tension, and C_{og} is the droplet field concentration, given by:

$$
C_{og} = \frac{M_{og}}{V_g} = \frac{\rho_{og} \alpha_{og}}{\alpha_{og} + \alpha_{gg}} \tag{156}
$$

The deposition velocity constant for horizontal flow ($k_{D,h}$) is set to 0.1 m/s. Both the entrainment and deposition correlations have been multiplied by the control volume length *L* and pipe perimeter *S* to get entrainment and deposition rates in the correct units (converting from mass rate per length and surface area kg/(m^{2}s) to kg/s).

2.10.2.2 Gas bubbles

The entrainment and deposition rates for gas bubbles in the oil film is presented here (the rates for gas bubbles in water are be calculated similarly). The entrainment rate of gas bubbles into the liquid film is by default zero. Most likely some entrainment will happen due to waves breaking, but currently the only way gas bubbles can enter into the liquid film is from the mass flux leaving a slug border (bubble-nose). Void in slug must in other words be enabled to obtain any gas in the liquid film in the bubble section.

For the deposition rate, we first need to find the terminal velocity of the gas bubbles in the liquid. This can be obtained from a force balance on a gas bubble between gravity and drag:

$$u_{T,go}^2 = \frac{4D_d g\left(\rho_{ol} - \rho_g\right)\cos\theta}{3C_D\rho_{ol}} \tag{157}$$

Here, D_d is the bubble diameter, and $u_{T,go}$ is the terminal velocity.

The drag coefficient can be estimated from the correlation of Clift and Gauvin [54] shown in equation (159), valid for $\mathrm{Re}_d < 3\cdot10^5$. In lack of a better alternative, equation (159) is also used above this range, but limited give a minimum value of 0.19. This is a value reported by Clift and Gauvin, which the drag coefficient approaches at high Reynolds numbers.

$$C_D = \max\left(24\frac{1+0.15\,\mathrm{Re}_d^{0.687}}{\mathrm{Re}_d}, 0.19\right) \tag{158}$$

The gas-bubble Reynolds number is defined by the bubble diameter and terminal velocity, and the continuous liquid density and viscosity:

$$\mathrm{Re}_d = \frac{\rho_{ol} u_T D_d}{\mu_{ol}} \tag{159}$$

Since there will be a distribution of bubble diameters of varying size, we use the Sauter mean droplet diameter ($D_d = d_{32}$) in equation (158). The Sauter mean diameter represents the size of the larger bubbles that contain most of the volume. This is in many correlations dependent on the energy dissipation, but for simplicity we use a more simplified model as presented in the TRAC-M theory manual [55] shown in equation (161). This might be a too simple model, and could be improved in further work.

$$\begin{aligned} d_{32} &= 2L_0 \\ L_0 &= \sqrt{\frac{\sigma_{gl}}{g\left(\rho_l - \rho_g\right)}} \end{aligned} \tag{160}$$

Together equations (158) - (161) gives a non-linear equation for the terminal velocity, which is solved by Newton's method. After the terminal velocity has been calculated, the mass rate of gas bubbles out of the liquid film is calculated from:

$$\phi_{d,go} = m_{go} A_{go} u_{T,go} = m_{go} S_{go} L u_{T,go} \tag{161}$$

Here A_{go} is the gas-oil interfacial area.

In vertical flow, the gravity no longer acts to make the bubbles rise out of the liquid film and into the gas. Instead, the gravity acts on the gas bubbles in the direction parallel to the flow, which has to be implemented by slip relations between the entrained field and the continuous liquid field. Consequently, the deposition rate of gas bubbles from the liquid film in vertical flow will be zero due to the cosine term in equation (158).

In vertical flow it might still happen that a lot of entrained gas accumulates in a liquid film, even if slip relations are applied. To avoid the gas in liquid fraction becoming too large, the following adjustment of the deposition rate is made:

$$\phi_{d,go} = \max\left({}_{go}A_{go}u_{T,go}, \max\left(\left(\frac{\alpha_{go}}{\alpha_{ol}} - 0.5 \right), 0 \right) \frac{M_{go}}{\Delta t} \right) \tag{162}$$

2.10.2.3 Slug front entrainment

Several correlations are available in the literature for the entrainment of gas into a slug front: Nydal and Andreussi [56], Ullmann and Brauner [57], Skartlien et al. [58], Manolis [59] and Chanson [60]. Based on relatively simple testing and from the results obtained by the tests performed by Bonizzi and Issa [61] a modified version of the correlation of Chanson was chosen. The model is modified with the inclusion of a critical Weber number as suggested by Ullmann and Brauner, and is shown in equation (164).

$$\begin{aligned}
&\tilde{\theta} = \begin{cases} |\theta| & |\theta| < 45 \\ 90 - |\theta| & |\theta| \geq 45 \end{cases} \\
&\tilde{d}_{cr} = \begin{cases} \dfrac{0.224}{\sqrt{\text{Eo} \cdot \cos\tilde{\theta}}} & \text{Eo} \geq 0.2 \\ 0.25 & \text{Eo} < 0.2 \end{cases} \\
&\text{We}_{cr} = \frac{100 \cdot 0.6}{\tilde{d}_{cr}} \\
&\text{We} = \rho_l \left(u_{l,front} - u_{l,bub} \right)^2 D / \sigma_{gl} \\
&\xi = \begin{cases} 1.245 & \text{Fr} < 7 \\ 1.4 & \text{Fr} \geq 7 \end{cases} \qquad \varsigma = \begin{cases} 0.018 & \text{Fr} < 7 \\ 0.014 & \text{Fr} \geq 7 \end{cases} \\
&\text{Fr} = \frac{\left| u_{l,front} - u_{l,bub} \right|}{\sqrt{gA_{l,bub} / S_{i,bub}}} \\
&\dot{M}_{slugfront} = \begin{cases} \rho_{gg,bub} A_{l,bub} \left| u_{l,front} - u_{l,bub} \right| \varsigma \max\left(\text{Fr} - 1, 0 \right)^{\xi} & \text{We} \geq \text{We}_{cr} \\ 0 & \text{We} < \text{We}_{cr} \end{cases}
\end{aligned} \tag{163}$$

Here Eo is the Eötvös number (see equation (183)), while We is the Weber number.
The entrainment rate is set to zero if the Froude number is less than one, or if the liquid in the film is not going into the slug front. Note also that the resulting entrainment rate was found to be too large if using the actual front velocity (border velocity at the slug front), while the results seemed more reasonable by using the liquid velocity at the slug front ($u_{l,front}$).

Note that void in slug can be enabled even if slug front entrainment is disabled, but gas can then only enter slugs either through mass sources, or from when converting a bubble to slug by distributing the gas present in the bubble as gas in the liquid slug.

2.10.3 Slip relations

Slip relations relate the velocity of the sub-fields to the mixture fields:

$$u_f^{n+1} = C_{f,slip}^n u_k^{n+1} + U_{f,slip}^n \tag{164}$$

By default no-slip is applied ($C_{f,slip} = 1$, $U_{f,slip} = 0$), except for the gas bubbles in liquid for which the Malnes slip relation [62] is default. Equation (166) shows the Malnes slip relation (here shown for the gas in oil field).

$$\begin{aligned} S_D &= \frac{1-\alpha_{go}}{0.95-\alpha_{go}} \\ C_{go,slip} &= S_D \\ U_{go,slip} &= S_D 1.18\left(\sigma_{go} g \frac{\rho_{ol}-\rho_{go}}{\rho_{ol}^2}\right)^{\frac{1}{4}} \left(\alpha_{ol}\sin\theta\right)^{\frac{1}{2}} \end{aligned} \tag{165}$$

2.11 Hydrodynamic slugs and slug initiation

How to initiate or capture hydrodynamic slugs is still one of the most challenging tasks in gas-liquid simulations with the two-fluid model. There are typically three different approaches:

1. Using the unit-cell model as a sub-grid model
2. Applying a fine grid to capture the natural growth of waves into slugs
3. Initiating slugs on a course grid using mechanistic initiation criteria

All three approaches can be used in the Sluggit framework, though they all have their shortcomings. The unit-cell model will only give the average friction terms to use in the momentum equation, treating the slug flow in a statistical manner and assuming steady state flow in each section. Information like slug length and slug frequency is thus not possible to obtain. The unit-cell model is described in section 2.15.

A promising approach is using so-called slug capturing, through the accurate numerical solution of the one-dimensional two-fluid model, typically employing higher order discretization schemes. This approach is believed to be capable of describing the transition from stratified flow to slug flow, see for instance the work by Issa [16, 63] and publications from the commercial simulator LedaFlow [9, 64, 65]. If the conditions are correct, small instabilities on the gas-liquid interface will become unstable and grow, either to stable waves, or to slugs. The downside of this approach is that using a fine grid means increased computational time. In addition, there are still some unresolved issues regarding the mathematical properties of the two-fluid model: The model can become ill-posed, which means that the solution will be dependent on the grid size. In opposite, a well-posed problem has the following properties:

1. A solution exists
2. The solution is unique
3. The solution's behaviour changes continuously with the initial conditions.

The characteristic roots of the mass and momentum equations shown in equation (167) gives the Inviscid Kelvin-Helmholtz limit [66] (also called the well-posedness criterion), the limit for when the eigenvalues for the system of equations becomes complex and the model is ill-posed. The model is then no longer hyperbolic, but becomes elliptic (or mixed hyperbolic/elliptic).

$$\left(u_G - u_L\right)^2 < \left(\frac{\alpha_L}{\rho_L} + \frac{\alpha_G}{\rho_G}\right)\frac{\rho_L - \rho_G}{\frac{\partial \alpha_l}{\partial h_l}} g\cos\left(\theta\right) \tag{166}$$

When the model becomes ill-posed, the growth rates increase exponentially as the wavelength shrinks to zero, which is unphysical. The reason for why the two-model can become ill-posed is that several terms are dropped in the averaging the full three-dimensional equation to the one-dimensional two-fluid model. The paper of Sanderse et al. [67] clearly shows how some of the skipped terms have to be reintroduced (re-introduced from data from CFD-simulations) to get the correct results for a relatively simple simulation of the drift velocity of a Taylor bubble in a channel. The commercial simulator LedaFlow used a different approach, and showed how the interfacial friction can be tuned to give the correct drift velocity [9]. The papers from Sanderse and LedaFlow are not directly related to the ill-posedness of the two-fluid model, but clearly illustrates that some basic physics has been lost in the averaging. Several attempts has been made to try to regularize the two-fluid model to cope with the ill-posedness, like adding numerical viscosity, surface tension and artificial interfacial pressure. For a review of such regularization techniques, see for instance the study of Fullmer [68]. Another approach is to employ a two-pressure model as investigated by among others Munkejord [69], but this method also seems to have its drawbacks. Though some of these attempts might help stabilize the two-fluid model, they all have their drawbacks, and some of the approaches even lack a good physical justification. In other words, the problem of an ill-posed the two-fluid model is yet to be solved.

Using a first order discretization for instance for the convective terms like the first order upwind scheme will yield a round-off error of similar form as when adding artificial numerical viscosity. This will stabilize the model, but also makes it more diffusive. Furthermore, one does not have control of what the numerical viscosity will be, and it will change with the grid size. One could in other words get similar results by using a higher order scheme and explicitly adding some numerical viscosity, using a larger grid size and thus also a larger time step.

It should also be noted that the two-fluid model is unconditionally ill-posed if the hydrostatic level gradient term is not included (which will always be the case in vertical flow).

The third possible way of dealing with hydrodynamic slugs is to use mechanistic initiation criteria. Two initiation criteria are implemented in Sluggit to insert slug section objects depending on the flow conditions.

One is based on the instability of stratified flow (equation (167)), and the other on the existence of slug flow derived by Bendiksen and Espedal [70] which is based on that the slug front velocity at steady state must be larger than the nose/tail velocity:

$$u_b < u_{g,bub} \frac{\alpha_{g,bub} - \alpha_{g,slug} \dfrac{u_{g,slug}}{u_{g,bub}}}{\alpha_{g,bub} - \alpha_{g,slug}} \tag{167}$$

Note that in the limit of zero gas in the slug, equation (168) reduces to $u_b < u_{g,bub}$. This criterion is currently only implemented to work with no void in slug. Note also that this is only a necessary criterion, but not a sufficient criterion for slug flow to occur. This criterion must in other words be used together with for instance the well-posedness criterion.

The slugs that are initiated are created in the middle of the bubble section, with a user-specified length in multiple of pipe diameters.

The commercial simulator OLGA has until recently initiated slugs in its slug tracking model from a user-specified delay constant and initiation frequency, which makes the result dependent on the user input and has been subject for criticism. Lawrence et al. [71, 72] however recently presented an improved slug initation model for OLGA, where the only required user input is the length of a short and stable slug. The initiation model is implemented as a conservation equation for the number of initiated slugs similar to the continuity equation, with the birth and death rate of the slugs as the source terms. The model is obviously dependent on good correlations for the birth rate and death rate, which are not given in full form in the papers. This type of initiation model has not yet been investigated in the Sluggit framework, but could be an alternative to investigate in future work.

2.12 Plug boundary movement

Slugs normally have a front and a tail, where liquid is absorbed at the front and shed at the tail. Hence, the velocities of the edges of a slug are generally not equal to the slug's bulk velocity. In this simplified approach, where slugs are confined within a single control volume, it is necessary to apply closure relations to determine the velocities of slug front and slug tail. (i.e. at the borders in Figure 5). The border velocities are expressed by a linear relationship with the liquid mixture velocity:

$$u_b = C_{0l} u_{l,j} + U_0 \tag{168}$$

The purpose of applying a moving grid is to make the grid coincide exactly with the edges of the slugs and plugs. Consequently, the border velocity coefficients are computed according to the relations supplied below.

The movement of the slug edges is related to the liquid velocity in the slug. If the slug did not lose or accumulate liquid in any way, the slug border velocity would simply equal the slug's liquid velocity.

2.12.1 Critical bubble turning velocity

In order to determine whether a slug border is a slug front or a bubble nose, one must apply some kind of bubble turning criterion. In the current work, the criterion is based on the assumption that bubbles tend to move in the opposite direction of the pressure gradient. This criterion has been shown to predict the bubble turning point quite well in [73, 74]. The bubble will thus turn when the gravitational forces balance the friction forces over the slug body:

$$\frac{0.125\rho_l \left|u_{crit}\right| u_{crit}\lambda_{wall}S_l}{A} = \left|\rho_l g \sin\theta\right| \tag{169}$$

Note that this expression gives the absolute value of the critical velocity, for simplicity. The correct sign will be applied after calculating the absolute value.

In laminar flow this equation has a simple analytical solution, by using the laminar friction factor from equation (146):

$$u_{crit,lam} = \frac{\rho_l g D^2 \left|\sin\theta\right|}{32\mu_l} \tag{170}$$

For turbulent flow however, when using the Haaland equation (equation (145)), we get a non-linear equation for the critical velocity:

$$u_{crit,tur} + 1.8\left(2gD\left|\sin\theta\right|\right)^{\frac{1}{2}} \log_{10}\left[\frac{6.9\mu_l}{\rho_l D u_{crit,tur}} + \left(\frac{\varepsilon / D}{3.7}\right)^{1.11}\right] = 0 \tag{171}$$

This can be easily solved with for instance Newton's method.

To determine if the laminar or turbulent critical velocity should be used, we first calculate the laminar critical velocity, and calculate the corresponding Reynolds number and friction factor. We then calculate the turbulent friction factor, using the critical velocity obtained from the laminar theory. If this turbulent friction factor is larger than the laminar friction factor, and the Reynolds number is above 300, we go on to solve equation (172). If not, we return the critical velocity calculated from the laminar theory.

Finally, the correct sign is of the critical velocity is applied by:

$$u_{crit} = \begin{cases} -u_{crit,abs} & \theta \geq 0 \\ u_{crit,abs} & \theta < 0 \end{cases} \tag{172}$$

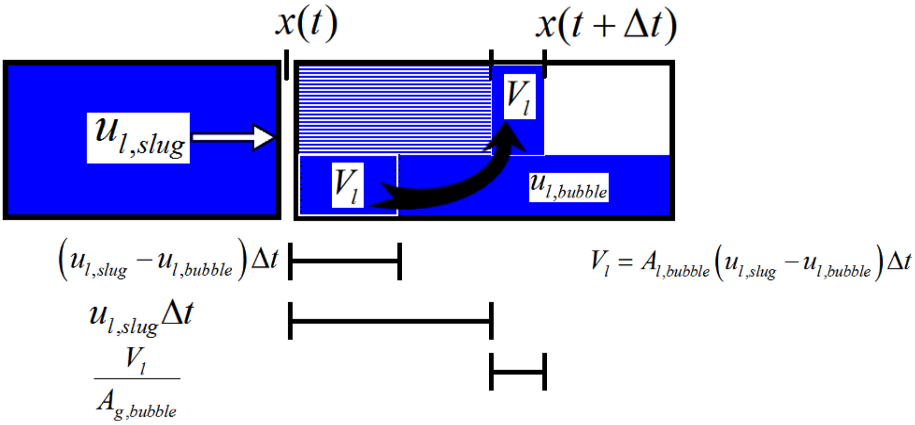
A slug border can then be determined to be a bubble nose or a slug front as shown in Table 4.

Table 4: How to determine if a slug border is a bubble nose.

	Slug-bubble	Bubble-slug
Nose criterion	$u_{crit} \geq u_l$	$u_{crit} < u_l$

2.12.2 Slug front velocity

For slug fronts, the liquid mass balance across the slug front must be fulfilled.



$x(t)$
$x(t+\Delta t)$
$u_{l,slug}$
V_l
V_l
$u_{l,bubble}$
$(u_{l,slug} - u_{l,bubble})\Delta t$
$u_{l,slug}\Delta t$
$\frac{V_l}{A_{g,bubble}}$
$V_l = A_{l,bubble}(u_{l,slug} - u_{l,bubble})\Delta t$

Figure 48: A slug front moving to the right, into the neighbouring bubble section.

During a time step Δt, the slug front will move a distance $u_{l,slug}\Delta t$ into the neighbouring bubble section. The slug front will then have absorbed the following liquid volume from the neighbouring bubble section:

$$V_l = (u_{l,slug} - u_{l,bubble}) A_{l,bubble} \Delta t \qquad (173)$$

The slug front must then be moved an additional distance to make room for this volume, as illustrated in Figure 48.

The velocity of the slug front is calculated by:

$$\begin{aligned} U_{front} &= u_{l,slug} + \frac{A_{l,bubble}}{A_{g,bubble}}\left(u_{l,slug} - u_{l,bubble}\right) \\ &= u_{l,slug} + \frac{\alpha_{l,bubble}}{\alpha_{g,bubble}}\left(u_{l,slug} - u_{l,bubble}\right) \\ &= u_{l,slug}\left(\frac{1}{1-\alpha_{l,bubble}}\right) + \frac{-u_{l,bubble}\alpha_{l,bubble}}{1-\alpha_{l,bubble}} \end{aligned} \tag{174}$$

Note that equation (175) is only used if liquid is going into the slug front. If the front is not absorbing liquid it is assumed to be a front moving with the mixture liquid velocity, and we set $C_{0l} = 1$ and $U_0 = 0$.

Note also that the velocity of the liquid going into the slug front is not taken as the liquid velocity in the bubble section next to the slug front. The velocity in this section is calculated as the average between the velocity at the slug border and the neighbouring bubble-bubble border, which means that this section velocity is coupled to the velocity in the slug. In reality, the liquid going into a slug front has no information about the velocity in the slug. Consequently the value for $u_{l,bubble}$ is taken from the mixture liquid velocity in the neighbouring bubble-bubble border.

Equation (175) is a linear relationship with the liquid mixture velocity in the slug, and we get the following coefficients:

$$\begin{aligned} U_{front} &= C_{0l}u_{l,slug} + U_0 \\ C_{0l} &= \frac{1}{1-\alpha_{l,b}} \\ U_0 &= \frac{-\alpha_{l,b}u_{l,bubble}}{1-\alpha_{l,b}} \end{aligned} \tag{175}$$

Note that the calculations above are only valid when there are no area changes. How the calculations are performed to account for changes in the cross-sectional pipe area is described in section 2.12.4.

When moving an incompressible slug front border, an additional movement Δx_{src} might be necessary to make room for the liquid that enter the slug from possible mass sources inside the slug:

$$\begin{aligned} \Delta V_{src} &= \sum_k \frac{\dot{M}_k^{src}}{\rho_k} \\ \Delta x_{src} &= \frac{\Delta V_{src}}{A} \end{aligned} \tag{176}$$

If there are changes in the cross-sectional area, the length needed to distribute the volume from the source is calculated numerically. This term is only included for incompressible slugs, since no pressure equation is solved. For compressible slugs the mass source terms enters as normal through the pressure equation, which again affects the new mixture liquid velocity and new border velocity.

It might happen that both the left and right slug borders are bubble-noses, in other words no slug fronts exist. The source term contribution will then be included for one of the bubble-nose borders.

2.12.3 Bubble nose velocity

A bubble nose velocity relation for inclined pipes has been proposed by Bendiksen [1] on the form:

$$u_b = C_{0l} u_l + U_0 \tag{177}$$

The drift velocity U_0 can be represented as split up into the vertical and horizontal components:

$$U_0 = U_{0v} + U_{0h} \tag{178}$$

A similar model to that of Bendiksen is implemented in the Sluggit code, on the form:

$$
\begin{aligned}
u_b &= C0_1 u_l + U_{0,v} + \mathrm{sgn}_h \left|U_{0,h}\right| && \mathrm{Fr} < 3.5 \\
u_b &= C0_2 u_l + U_{0,v} && \mathrm{Fr} \geq 3.5 \\
C0_1 &= \left(C_{0,1} + 0.15\sin^2(\theta)\right) \\
C0_2 &= C_{0,2} \\
U_{0,v} &= F_{rv}\sin(\theta)\sqrt{\frac{\left(\rho_l - \rho_g\right)}{\rho_l} gD} \\
U_{0,h} &= F_{rh}\cos(\theta)\sqrt{\frac{\left(\rho_l - \rho_g\right)}{\rho_l} gD}
\end{aligned}
\tag{179}
$$

The coefficients $C_{0,1}$ and $C_{0,2}$ varies depending on the conditions of the flow, and are described in more detail further below. The variable Fr is the Froude number, which is the ratio of inertia to gravitational forces:

$$
\mathrm{Fr} = u_l\sqrt{\frac{\rho_l}{gD\left(\rho_l - \rho_g\right)}} \tag{180}
$$

The parameter sgn_h is +1 or -1, depending on the orientation of the bubble-nose. If the nose is pointing left it is -1, and +1 if the nose is pointing to the right. The sign of the vertical drift velocity is automatically correct since it is multiplied by $\sin(\theta)$ (for inclinations within $\pm 90\,^{\circ}$).

The Taylor-bubble tends to travel along the top of the pipe, instead of being symmetric around the pipe centre line. This effect is represented by the first equation in equation (180), where the Froude number is low. When the velocity is increased, the bubble nose is forced towards the centre of the pipe, and the second equation in equation (180) is used.

Instead of using the Froude number directly to calculate the transition from the low Froude number to the high Froude number region, the velocity at which the transition occur can be calculated by:

$$u_{l,lim} = \left| \frac{U_{0,h}}{C0_2 - C0_1} \right| \qquad (181)$$

If the absolute value of the liquid mixture velocity is larger than $u_{l,lim}$, the high Froude number coefficient is used.

The coefficients F_{rh} and F_{rv} in equation (180) are related to the horizontal and vertical drift velocity of the Taylor bubble. Benjamin [2] calculated the coefficient for the horizontal drift velocity (F_{rh}) to be 0.542, but this calculation does not take into account the liquid viscosity. Jeyachandra et al [4] performed experiments using different liquid viscosities, and proposed a viscosity dependent correlation for the horizontal drift velocity coefficient, which is used in the Sluggit framework:

$$\begin{aligned} \mathrm{Ar} &= \frac{D^{1.5}\sqrt{\rho_l(\rho_l - \rho_g)g}}{\mu_l} \\ \mathrm{Eo} &= \frac{\left(\rho_l - \rho_g\right)gD^2}{\sigma_{gl}} \\ F_{rh} &= 0.53e^{\left(-13.7 Ar^{-0.46} E\ddot{o}^{-0.1}\right)} \end{aligned} \qquad (182)$$

Ar and Eo are the Archimedes and Eötvös numbers respectively, describing the ratio of gravitational to viscous forces, and the ratio of buoyancy to surface tension forces. Dumitrescu [3] computed the approximate shape of a bubble rising in a vertical tube from theoretical considerations. By ignoring the frictional and capillary effects, and by considering only the potential and kinetic energy of the liquid falling around the bubble, the approximate solution for the liquid flow around the top of the bubble was determined by both Dumitrescu and Davies & Taylor. The solution published by Dumitrescu gives a value of 0.351 for the vertical drift velocity coefficient, a value that has been widely used.

Joseph [5, 6] presented an analytical expression for the rise velocity of a spherical cap bubble to account for the effect of liquid viscosity, which we used for the vertical drift velocity coefficient in the Sluggit code:

$$F_{rv} = -\frac{8}{3}\frac{\mu_l}{\rho_l}\frac{(1+8s)}{\sqrt{gD^3}} + \frac{\sqrt{2}}{3}\left[1-2s-\frac{16s\sigma_{gl}}{\rho_l gD^2}+\frac{32\mu_l^2}{\rho_l^2 gD^3}(1+8s)\right]^{0.5} \tag{183}$$

Here, *s* is a parameter describing the deviation of the free surface from perfect sphericity, in other words related to the shape of the bubble. The bubble nose is more pointed when s < 0 and blunted when s > 0. This factor is not known, and in the Sluggit code we use s=0.2228 which will give approximately the value of Dumitrescu for zero viscosity (as long as the pipe diameter is not too small).

The parameters $C_{0,1}$ and $C_{0,2}$ in equation (180) captures the effect of the velocity profile, in which the fluid velocity at the centre of the pipe is greater than the average velocity, making the bubble front propagate faster than the mixture. To calculate these coefficients we use a method proposed by Nuland [75]. We define Re_{tr} as the transition Reynolds number, calculated from where the maximum value of the laminar and turbulent friction factors gives a transition from laminar to turbulent flow. The Reynolds number Re_t gives the limit for turbulent flow and is set to 3000, while Re_{slug} is the Reynolds number in the slug. For $\mathrm{Re}_{slug}<\mathrm{Re}_{tr}$ we assume laminar flow, and set $C_{0,1}$ and $C_{0,2}$ equal to 2.

Chen [76] showed that for turbulent flows ($\mathrm{Re}_{slug}>\mathrm{Re}_{tr}$), the ratio between the maximum velocity and the mean velocity in single phase flow can be calculated as:

$$\begin{aligned} C_{0tur} &= \frac{(n+1)(2n+1)}{2n^2} \\ n &= \kappa\sqrt{\frac{8}{f}} \end{aligned} \tag{184}$$

The parameter *n* represents the inverse exponent in the power law velocity profile:

$$u = u_{\max}\cdot\left(\frac{R-r}{R}\right)^{\frac{1}{n}} \tag{185}$$

The friction factor f is the turbulent friction factor using the Darcy friction factor formulation, while κ is the von Karman constant (=0.41). We use the friction factor in the slug for *f*. This expression is valid for both hydraulic smooth and rough regimes. In the transitional region between laminar and turbulent flow (Re$_{tr}$ <Re$_{slug}$<Re$_{tur}$) we use the following interpolation method:

$$\begin{aligned} w &= \frac{\mathrm{Re}_{slug} - \mathrm{Re}_{tr}}{\mathrm{Re}_{tur} - \mathrm{Re}_{tr}} \\ C_{0tr} &= wC_{0tur} + (1-w)C_{0lam} \end{aligned} \tag{186}$$

The value of C_{0lam} is set to 2. It should be noted that according to Wallis [77], C_{0lam} can be even higher than 2 for fully laminar flow, but the exact behaviour has yet to be determined. The user can specify both the value of C_{0lam}, and the transitional and fully turbulent Reynolds numbers. The fully turbulent values of $C_{0,1}$ and $C_{0,2}$ (C_{0tur}) are 1.05 and 1.2, respectively, and the calculation of the transitional coefficient C_{0tr} is limited to not be smaller than this.

If however the bubble motion is in the opposite direction of the flow, the mixture velocity profile is directed in the opposite direction of the bubble nose, and a lower value for $C_{0,1}$ and $C_{0,2}$ is expected (referred to here as $C_{0,counter-current}$). This will happen for the following conditions:

1. For a slug-bubble border that has been determined to be a bubble-nose and u_l is larger than zero.
2. For a bubble-slug border that has been determined to be a bubble-nose, and u_l is less than zero.

Experimental results [78] suggest a value near 1 in such cases. The default value in Sluggit is set to 0.98.

To sum up, when the coefficients has been calculated the procedure is as shown in Table 5.

Table 5: Summary of cases in nose-velocity calculation.

Condition	Case	Coefficients
Left pointing nose and $u_l > 0$ or right pointing nose and $u_l < 0$	Counter-current flow	$C_{0l} = C_{0,counter-current}$ $U_0 = U_{0,v} + \mathrm{sgn}_h \left\| U_{0,h} \right\|$
$\left\|u_l\right\| < \left\|u_{l,lim}\right\|$	Low Froude number case	$C_{0l} = C0_1$ $U_0 = U_{0,v} + \mathrm{sgn}_h \left\| U_{0,h} \right\|$
$\left\|u_l\right\| \geq \left\|u_{l,lim}\right\|$	High Froude number case	$C_{0l} = C0_2$ $U_0 = U_{0,v}$

Note that the calculations above are only valid when there are no area changes. How the calculations are performed to account for changes in the cross-sectional pipe area is described in section 2.12.4.

2.12.3.1 Wake effect

It is also possible to include a wake effect on the bubble-nose velocity. The C_{0l} and U_0 coefficients of the bubble-nose border will then be multiplied by the wake effect factor *W*. Currently the correlation of Moissis and Griffith [79] is implemented, where *L* is the length of the slug unit and *D* is the pipe diameter:

$$W = \left(1 + 8e^{-1.06\frac{L}{D}}\right) \tag{187}$$

2.12.4 Plug-plug border velocities

For plug-plug borders, the border coefficients C_{0l} and U_0 are set to 1 and 0 respectively, so that the border velocity will equal the mixture liquid velocity. This will be the case for both slug-slug borders, pig-pig borders, slug-pig borders and pig-slug borders, for both compressible and incompressible slugs.

2.12.5 Incompressible plug unit velocities

As mentioned in section 2.3, the velocities are solved for the slug unit for incompressible slugs, and not for the slug borders. If the slug unit consists of different cross sectional areas, the velocity we solve for is defined through the average area of the slug unit ($\frac{V}{L}$).

Since the slugs are incompressible, we can calculate the volume flow through the slug unit as:

$$Q_{l,slug} = \frac{V_{slug}}{L_{slug}} u_{l,slug} = \overline{A}_{slug} u_{l,slug} \tag{188}$$

The velocity at any point in the slug unit can then be calculated by using the local area at the location we want the velocity:

$$u_{l,local} = \frac{Q_{l,slug}}{A_{local}} \tag{189}$$

If the cross sectional area is the same everywhere in the slug, the velocities are the same at all locations in the slug. See also section 2.9.1.3.

2.12.6 Plug movement and area changes

After solving the new velocities in the slug and calculating the new border velocities, the borders are moved a distance Δx :

$$\Delta x = u_b^{n+1} \Delta t \tag{190}$$

This is however only valid when there are no area changes. The new border velocity u_b^{n+1} is the new velocity at the new position, and not the average velocity the borders moves by from time *n* to *n*+1. If there are area changes in the region the border moves, the mixture liquid velocity in the slug and thus also the border velocity might have different values at the different cross sectional areas in the region the border is going to move.

Sections 2.12.6.1 and 2.12.6.2 describe how the border movement should be performed to account for area changes for a bubble-nose and a slug front respectively. Common for both the movement of the bubble-nose and the slug front is that the volume flux in/out of the border is stored, and used directly in the momentum and mass equations instead of the regular flux terms. The stored volume flux is named $Q_{k,slug,stored}$. That is, instead of calculating the mass flux from $m_k A_k (u - u_b)$, we calculate the mass flux as $m_k Q_{k,slug,stored}$. The stored fluxes are updated in each volume fraction iteration. In the pressure equation however we want to keep the original formulation of the mass fluxes and relate them to the change in velocity to add implicitness to the momentum equation. If there are area changes, the stored flux will typically be different than the original flux formulation, so that a set of scaling constants are needed in the pressure equation (described in section 2.9.4.4).

2.12.6.1 Bubble-nose movement and area changes

The velocity at the bubble-nose is defined at the old area (A^n), where the border velocity coefficients were calculated. To be able to calculate how far the border should move, we start by calculating the expected volume the border should pass by. This can be done by first calculating the new mixture velocities in the slug, but using the old cross sectional area where the velocity was defined, and then calculate a new temporary border velocity from this:

$$\begin{aligned} u_l^* &= \frac{Q_{l,slug}^{n+1}}{A^n} \\ u_b^* &= u_l^* C_l + U_0 \end{aligned} \tag{191}$$

The expected volume this border should move past will then be:

$$V^* = u_b^* A^n \Delta t \tag{192}$$

The length (in the correct direction) from the old position that corresponds to this volume is then calculated numerically, and this will be the length this border should move. If the cross sectional areas are the same everywhere, this result will be identical to equation (191).
The volume flux at the bubble-nose border that will be stored for usage in the mass, momentum and energy equations will be:

$$Q_{l,slug,stored} = A^n \left(u_l^* - u_b^* \right) \tag{193}$$

The new border velocity defined at the new location of the border must be scaled by the ratio of the new area to the old area for the bubble-nose (this is not necessary for the slug-front). Since the liquid velocity is already scaled (see equation (190)), only U_0 is left to scale. The new border velocity for the bubble-nose can thus be calculated as:

$$u_{bubble-nose}^{n+1} = C_{0l}^n u_l^{n+1} + \frac{A_j^{n+1}}{A_j^n} U_{0,j}^n \tag{194}$$

2.12.6.2 Slug front movement and area changes

A similar approach as for the bubble-nose is used, where we try to calculate the expected volume the border should move past and then find the corresponding length. The calculation for the slug front however becomes more difficult, because the different parts of C_l and U_0 that comes from how much liquid the front will shed or take up will be different for different cross sectional areas (see section 0). To get this correct, we calculate and sum up the different contributions from the volumes illustrated in Figure 48 in individual steps numerically instead of simplifying them analytically into the C_l and U_0 coefficients. Note that these volumes will have a sign, so that some volumes are negative.

First we calculate the volume the slug front will move past if it moves with the new liquid velocity, calculated using the old area:

$$V_{front,1} = u_l^* A^n \Delta t \tag{195}$$

Next the length (in the direction u_l^* goes) that gives the volume $V_{front,1}$ is calculated numerically, which we name $\Delta x_{front,1}$. This is the length the front would have moved if it moves with the new liquid velocity, and we forget about there being any liquid in the neighbouring bubble section. The next step is to calculate the liquid exchange with the neighbouring bubble, taking into account that the liquid in the bubble is also moving. Two volumes are then needed. The first we call $V_{front,2}$, and is the liquid volume exchange between the slug front and the bubble if we assume that the liquid in the bubble is not moving. This volume is calculated numerically from position x_{front}^n to $x_{front}^n + \Delta x_{front,1}$ and is assigned the same sign as the sign of $V_{front,1}$. This corresponds to the term $A_{l,bubble} u_{l,slug} \Delta t$ in equation (174). Note that the positions, holdup and other values used in these calculations should be from time step *n*, and not new values. This is because C_l and U_0 are defined using old values, and only the new liquid velocity should be from time step *n*+1. The next term is also related to the liquid exchange between the slug front and the bubble, and comes from that the liquid in the bubble might be moving. This corresponds to the term $A_{l,bubble}u_{l,bubble}\Delta t$ in equation (174). Since this volume calculation is performed in the same region as $V_{front,2}$, we can calculate it as:

$$V_{front,3} = -\frac{V_{front,3}}{\Delta x_{front,1}} u_{l,bubble}\Delta t \tag{196}$$

The net volume exchange over the slug front border can then be found by summing up the three calculated volumes:

$$V_{front} = V_{front,1} + V_{front,2} + V_{front,3} \tag{197}$$

To make room for this volume, it is distributed in the gas region in the neighbouring bubble section. This is also calculated numerically, and gives the final length the front should be moved which we name $\Delta x_{front,2}$. This calculation corresponds to the term $\frac{V_l}{A_{g,bubble}}$ in Figure 48. The total length the slug front should move can then be calculated as $\Delta x_{front} = \Delta x_{front,1} + \Delta x_{front,2}$.

The volume that will be stored as the volume flux through the slug front can now be calculated as the total volume from position $x^n_{front} + \Delta x_{front,1}$ to $x^n_{front} + \Delta x_{front}$.

This procedure gives the same result as equation (191) if there are no changes in cross sectional area (this has been tested).

2.12.6.3 Example of plug movement and area changes

It is easy to see how the plug border movement would become wrong if not using the proposed procedure. In this simple example an incompressible slug is in contact with the inlet mass flow boundary at time t=0 seconds, with the front of the slug located at 0.499999 m. The geometry consists of two pipes of length 0.5 m, with pipe diameters 0.1 and 0.05 m which results in a four times smaller cross sectional area for the second pipe. These two cross sectional area will be referred to as A_1 and A_2 respectively. The liquid density equals 1000 kg/m^3 and is constant, and the water mass flow is set to give a velocity of 1 m/s at the first cross sectional area (approximately 7.854 kg/s). This gives a volume flow $\dot{Q}_l$ of 0.007854 m^3/s. Since this is a slug front not absorbing liquid, the border velocity coefficients C_l and U_0 becomes 1 and 0 respectively.

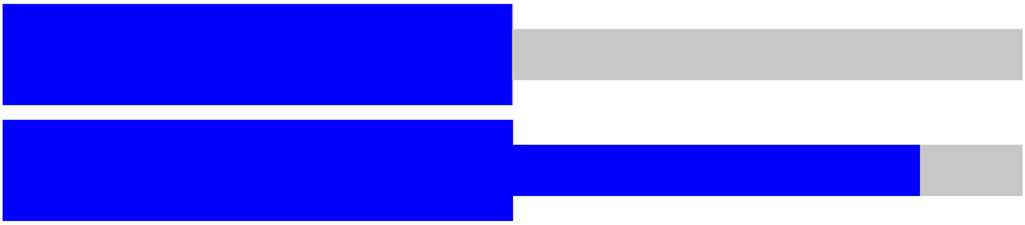


Figure 49: An incompressible slug moving from one cross sectional area into a four times smaller area. The top and bottom images correspond to times 0 and 0.1 seconds respectively. In the bottom image the slug front has moved a distance 0.4 m to the right.

The time step Δt in the simulation is set to 0.1 s, which means that a volume of $\dot{Q}_l \Delta t$ = 0.0007854 m^3 will enter the pipe during each time step. If we define the new border velocity at the cross sectional area at time step *n*, the new border velocity u_b would equal the new liquid velocity u_l at this location (since $C_l = 1$ and $U_0 = 0$), which would be $u_l = \frac{\dot{Q}}{A_1}$ = 1 m/s. This wold result in the border moving a distance $\Delta x = u_b \Delta t$ = 0.1 m, which would be wrong and yield a volume fraction error of 14.3 % in the slug. The correct result would be approximately $\Delta x = \frac{\dot{Q}\Delta t}{A_2}$ = 0.4 m. Defining the border velocity at the cross sectional area at time step *n*+1 would give an approximately correct result for this case, but would in many other cases not be correct. For instance in the case where the slug front starts at position 0.40001 m, and ends up at approximately position 0.50001 m. It is in simply not possible to express the distance the border should move as function of a border velocity defined at a single point. Instead, the volume the plug borders should cover are calculated, and the distance corresponding to these volumes numerically calculated as described in sections 2.12.6.1and 2.12.6.2. In addition, the correct scaling constants are applied to transform the border velocity used in the pressure equation from a locally defined velocity to the average border velocity between times *n* and *n*+1.

2.12.7 Pressure gradient over plugs

As shown in Figure 5, Figure 6 and Figure 50, the momentum control volume of the plugs is non-staggered and does not extend into the neighbouring bubble sections. Consequently, the pressure is no longer defined at the end-point of the plug momentum control volume (at the plug borders). The same will be true for the pressure at the end-point of the momentum control volume of the neighbouring bubble-bubble border, which is extended all the way to the plug.

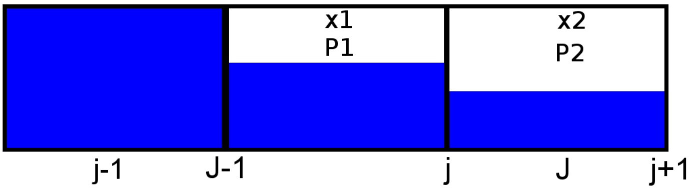


Figure 50: Indexing and pressures for the momentum equations next to a plug.

We will here look at an example for the right border of slug.
The default approach is to approximate the pressure at the plug border (index *J*-1) by the pressure in the bubble section (*P*1), but it is also possible to approximate the pressure by extrapolation assuming a linear pressure gradient from pressure P2 to P1.

The extrapolation for the pressure at index *J*-1, used both for the pressure gradient over the slug at index *j*-1 and for the bubble-bubble border at index *j*, becomes:

$$
\begin{aligned}
&p_{J-1} = p_1 - \frac{p_2 - p_1}{0.5\left(x_1 + x_2\right)} 0.5 x_1 \\
&\frac{\partial p_j}{\partial x} = \frac{p_J - p_{J-1}}{\left(x_1 + 0.5 x_2\right)} = \frac{p_2 - p_1 - \dfrac{p_2 - p_1}{0.5\left(x_1 + x_2\right)} 0.5 x_1}{\left(x_1 + 0.5 x_2\right)} = \frac{p_2 - p_1}{\left(x_1 + 0.5 x_2\right)} \left(1 - \frac{0.5 x_1}{0.5\left(x_1 + x_2\right)}\right)
\end{aligned}
\tag{198}
$$

New values (from time step *n*+1) are used for the both the pressure and lengths in equation (199), which are updated after each iteration on the system of equations.

The same interpolation can also be performed for the momentum control volumes next to a closed border since the control volume extent is modified also there. After solving the system of momentum equations the estimated pressure gradient is compared to the actual pressure gradient, and is said to be converged if the absolute value of the difference is less than the user-specified tolerance (default value 5 Pa/m).

Testing has however shown that including this interpolation does not give much difference in the results, though this might be case dependent. The cases marked as "case B" in Paper 1 were simulated with and without the pressure extrapolation enabled, and the slugging periods did not change much. The pressure signal at the start of the pipeline is plotted for a selected case in Figure 51, and it can be seen that the difference between the two simulations is quite small. The simulation run-time is however about 30 % larger when including the pressure extrapolation, as the pressure interpolation becomes a part of the iteration procedure to solve the system of equations.

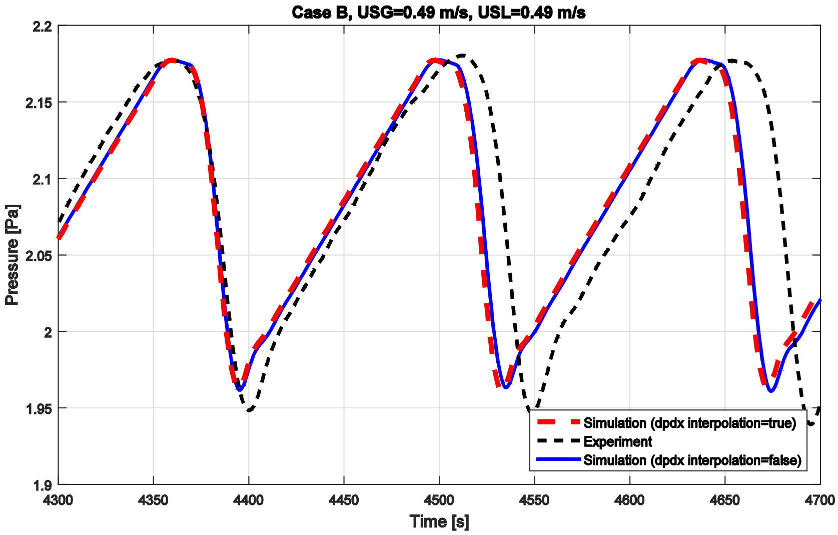


Figure 51: Simulated and experimental pressure for a selected experiment from Paper 1, showing simulations both with and without pressure extrapolation at closed borders and at slug borders.

2.13 Interfacial mass transfer

The interfacial mass-transfer model applied in Sluggit is the same as described in [80]. This model can treat both normal condensation or evaporation and retrograde condensation, in which a heavy phase condenses from the gas phase as the pressure drops. By defining the mass fraction at equilibrium as:

$$R_{s,k} = \frac{m_k}{m_g + m_l} \tag{199}$$

, the mass transfer rate for field *f* may be computed as:

$$\frac{dm_f}{dt} = \left[\left(\frac{\partial R_{s,f}}{\partial p} \right)_T \frac{dp}{dt} + \left(\frac{\partial R_{s,f}}{\partial p} \right)_T \frac{\partial p}{\partial x} \frac{\partial x}{\partial t} + \left(\frac{\partial R_{s,f}}{\partial T} \right)_p \frac{dT}{dt} + \left(\frac{\partial R_{s,f}}{\partial T} \right)_p \frac{\partial T}{\partial x} \frac{\partial x}{\partial t} \right] \left(m_g + m_l \right) \tag{200}$$

The term $\left(\frac{\partial R_s}{\partial p}\right)_T \frac{dp}{dt}$ represents the phase transfer from a mass present in a section owing to pressure change in that section, while the term $\left(\frac{\partial R_s}{\partial p}\right)_T \frac{\partial p}{\partial x}\frac{\partial x}{\partial t}$ represents the mass transfer caused by mass flowing from one section to the next. The gas mass fraction at equilibrium (R_s) must be specified, from a PVT-file.

Including interfacial mass transfer in the simulation is currently only available as a part of the volume fraction error iteration procedure, due to the way the computed mass changes are handled which will soon be described. Thus the effect on interfacial mass transfer from the energy equations will only be included if the energy equations are also a part of the volume fraction error iteration procedure. In addition, a minimum of two volume fraction error iterations must be used.

The reason for this is that after mass, pressure, momentum and energy has been solved, the interfacial mass transfer is computed and stored in the same arrays as contains the mass source terms. The mass, pressure, momentum and energy equations will thus not feel the effect from interfacial mass transfer until the next iteration, where it will enter through the source terms. At the end of iteration nr. 2 new values for the interfacial mass transfer will be computed and replace the current interfacial mass transfer values in the array containing the source terms, and this procedure is repeated until convergence. The iteration procedure will then converge the interfacial mass transfer as an implicit contribution.

Interfacial mass transfer will occur because a lump of fluid experience a change in pressure and temperature (or just one of them). For a specific section, this effect is computed both for fluids that remains in the section, enters the section from the left, or enter the section from the right. In addition, the effect is included for all types of source terms entering the section, from both mass sources and from other physical processes that are treated like source terms. This could be entrainment or deposition, slug front entrainment, slug-tail bubble-disengagement, and bubble-disengagement from the liquid film. The interfacial mass transfer for a sub-field phase f can then be calculated as follows:

$$dM_{f,IFM} = M_f\left[R_s\left(p_{new},T_{new}\right) - R_s\left(p_{old},T_{old}\right)\right] \tag{201}$$

Here M_f represents the mass of the lump of fluid that experience a change in pressure and temperature from p_{old} and T_{old} to p_{new} and T_{new}, while $dM_{f,IFM}$ represents the resulting mass transfer in kg.

Note that if the field is a gas field (gas in oil, gas in water or gas in mixture gas), a positive change in gas mass fraction (R_s) would mean gas turning into gas. This would in other words not represent any interfacial mass transfer, so for the result to be used $dM_{f,IFM}$ must be negative for the gas phase. Similarly, the mass transfer must be positive for liquid.

Gas mass originating from interfacial mass transfer in a slug section is put as gas in liquid/void fraction in the slug. An overview of the field that will receive mass from another field during an interfacial mass transfer process is shown in Table 6.

Note that the interfacial mass transfer is currently computed assuming two-phase flow, and will not work if both gas, oil and water are present. This would require that the PVT-file also contain information about the water mass fraction.

Table 6: Overview of the receiving field in an evaporation or condensation process.

Evaporating or condensating field	Receiving field
GG	OL or WL
OG	GG
WG	GG
OL	GG (GO for slugs)
WL	GG (GW for slugs)
GO	OL
GW	WL

2.14 Computational sequence

The computational sequence in Sluggit is as follows:

List 1: Computational sequence in the Sluggit code

1. Calculate and store values needed for iteration procedure
2. Solve equations with iteration procedure
3. Grid management

First, the current values for most parameters are stored as backup in all of the border and section objects. Then the border velocity and slip relation coefficients are computed, and all of the explicitly treated terms are calculated and stored to avoid unnecessary re-calculations of these (see the momentum equation in equation (88)).

2.14.1 Solving the system of equations

The iteration procedure to solve the system of equations is as follows:

List 2: Default solution procedure for the system of equations

1. Insert pressure equation into momentum equation and solve new velocities
2. Insert new velocities into pressure equation and calculate new pressure
3. Move borders
4. Solve mass equation and update holdup
5. Solve energy equation (if specified as a part of the iteration procedure)
6. Update densities, other physical quantities and holdup
7. Calculate and store interfacial mass transfer
8. If a fully implicit simulation is desired, re-calculate the necessary terms that are treated explicitly in the default scheme and stored before the iteration procedure
9. Repeat until convergence, using new values for the non-linear terms

Depending on the settings, the energy equation is solved as a part of the iteration procedure, and if the simulation is specified as not being isothermal. If it is specified not to iterate on the energy equations, they will be solved after the iteration procedure is finished. The effect of temperature will then be felt by the system through the volume fraction error in the pressure equation in the next time step.

At the end of each iteration a set of stopping criteria are checked:

List 3: Convergence criteria when solving the system of equations

1. The maximum volume fraction error in the system must be smaller than the user-specified tolerance
2. The maximum equation residual from the pressure and momentum equations must be below the limits given by the user (the default is a huge number to disable this check and only use the volume fraction error)
3. Other special criteria, like convergence of the volume flow in incompressible plugs which can be an issue for explicitly updated slugs with changes in cross-sectional area. If specified to do pressure gradient extrapolation as described in section 2.12.7 this is also checked to converge.

If all of the stopping criteria are satisfied, or the maximum number of allowed iterations is reached, the iteration procedure is stopped.

Note that the residuals should be computed after all of the different equation systems have been solved. If the iteration procedure is not able to converge, the time step is re-evaluated using a lower value for Δt .

It is also implemented to solve the mass, momentum and pressure equations all at once using Newton's method, which in certain cases can be significantly faster (for instance in the periodic Kelvin-Helmholtz simulations presented in Paper 2). This depends most likely on both the case itself and the limit for the convergence criteria.
The energy equation will however not be included together with mass, momentum and pressure, and is solved separately as normal.

2.14.2 Grid management and time step control

After the system of equations has been successfully solved, the next thing to do is to convert, remove or split sections. Long sections are split into smaller sections depending on user specified criteria, and sections might be removed based on different rules. They can be merged with similar sections if the length is too small, or converted from one section type to another for instance if the holdup in a bubble has become so large that it should be converted to a slug. This is all described in more detail in section 2.15.

The total mass of each phase in the pipeline is now stored, and will be used to check the mass conservation in the next time step. The results are now written to the output files, depending on the write interval. Next it is checked if any of the commands specified in the input file should be executed.

Finally the next value for the next time step Δt is calculated from the CFL criterion, as described in section 2.4. Depending on the interval specified in the input file, the current state of the entire simulation is dumped to a binary restart file, which can be used to restart the simulation from this point in time.

2.15 Unit-cell model

A unit-cell model has been implemented in the Sluggit framework, which can be used as a sub-grid model when using a large grid, or compiled separately and used as a stand-alone point-model. The concept of a unit-cell model was initially introduced by Dukler and Hubbard, see [14]. The current model is inspired by the work in Smith et al [81].

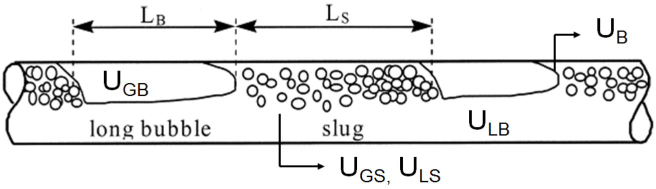


Figure 52: Schematic description of slug flow in a circular pipe.

The unit-cell model assumes fully developed flow, a train of Taylor-bubbles and slugs in a periodic frame of reference moving with the bubble nose velocity. The velocities and densities of each phase are assumed constant along the pipe, which gives the following equations for the volume flux:

$$\begin{aligned} \alpha_s U_{gs} + \left(1-\alpha_s\right) U_{ls} &= U_m \\ \alpha_b U_{gb} + \left(1-\alpha_b\right) U_{lb} &= U_m \end{aligned} \tag{202}$$

Here α_s and α_b are the void fractions in the slug and bubble regions, U_{ls} and U_{gs} are liquid and gas velocities in the slug, and U_{lb} and U_{gb} are liquid and gas velocities in the bubble region. In addition, continuity of the volume flux across the Taylor bubble nose gives:

$$\begin{aligned} \alpha_s (U_{gs} - U_b) &= \alpha_b (U_{gb} - U_b) \\ (1-\alpha_s)(U_{ls} - U_b) &= (1-\alpha_b)(U_{lb} - U_b) \end{aligned} \tag{203}$$

Combining equations (203) and (204) one gets the following equation for the superficial velocities in the bubble region and slug region:

$$\begin{aligned} U_{sgs} &= \alpha_s U_m \\ U_{sls} &= \left(1-\alpha_s\right) U_m \\ U_{sgb} &= U_{sgs} + U_b \left(\alpha_b - \alpha_s\right) \\ U_{slb} &= U_m - U_{sgb} \end{aligned} \tag{204}$$

The slug fraction *SF* which describes the ratio of the length of the slug region to the bubble region and can be calculated by:

$$SF = \frac{U_{sg} - U_{sgb}}{U_{sgs} - U_{sgb}} \tag{205}$$

A slug fraction between 0 and 1 corresponds to slug flow, while a slug fraction larger than 1 corresponds to bubbly-flow (no bubble-region). Stratified flow corresponds to a slug fraction less than 1. At steady state conditions, the following momentum balance for the gas and liquid is obtained in the bubble-region:

$$\begin{aligned} A_l \frac{\partial p}{\partial x} + \frac{A_l \rho_l K |u_l| u_l}{2\Delta x} + \tau_{wl} S_l - \tau_i S_i - \rho_l A_l g \sin\theta = 0 \\ A_g \frac{\partial p}{\partial x} + \frac{A_g \rho_g K |u_g| u_g}{2\Delta x} + \tau_{wg} S_g + \tau_i S_i - \rho_g A_g g \sin\theta = 0 \end{aligned} \tag{206}$$

Here the effect from losses due to for instance valves and expansion/contraction losses has also been added (summing up all K-factors). The term with the K-factor includes a length Δx, which is the length of the section we solve the unit-cell for ($\frac{K}{\Delta x}$ will then be an additional input to the point model).

Combining the two momentum equations and eliminating the pressure gradient, the following implicit equation for the liquid holdup in the bubble region is obtained:

$$f(H_l) = \tau_{wg} \frac{S_g}{A_g} - \tau_{wl} \frac{S_l}{A_l} + \tau_i S_i \left(\frac{1}{A_l} + \frac{1}{A_g} \right) - \left(\rho_l - \rho_g \right) g \sin\theta + \frac{K}{2\Delta x} \left(\rho_g |u_g| u_g - \rho_l |u_l| u_l \right) = 0$$

(207)

For consistency, the same friction models as are used in the dynamic model are used in the point model framework. Certain additional correlations are needed in the steady state point model: The entrained droplet fraction and the slug void fraction at steady state. In the dynamic model the entrainment mass *rates* are needed, while the point model needs steady state values. For consistency, the values at steady state given by the correlations used in the point model should match the values one would get when reaching steady state in a dynamic simulation. Since the entrainment and deposition rates of droplet entrainment are equal at steady state conditions, one possibility is to use the equations for the entrainment and deposition rates and solve for the entrained fraction that gives equal rates (the same applies for flux of gas entrainment in and out of the slugs). This was tested for the droplet entrainment and deposition rate correlation of Govan [53] described in section 2.10, but proved to be difficult. It was found that the correlations of Govan has multiple solutions, and in certain straight pipe test cases with dynamic simulations the entrained fraction was fluctuating between two steady state values. Consequently the point-model does not yet work with entrainment enabled. In addition, for void in slug the correlations used in the point model will not give the same steady state value as what will be reached at steady state conditions in a dynamic simulation. Instead the void in slug is calculated from the correlation of Gregory [82], shown in equation (209). Note also that no-slip is assumed in the point model.

$$\alpha_s = 1 - \frac{1}{1 + \left(\frac{U_m}{8.66} \right)^{1.39}} \tag{208}$$

Here U_m is the mixture velocity of gas and liquid.

Getting the steady state entrainment fraction and slug void fraction consistent with the dynamic simulations has simply not been a priority, as the point-model is used mainly as a steady state solver to be able to start a dynamic simulation from steady state conditions (see section 2.16 regarding the steady state solver). It is however also possible to use the point-model as a sub-grid model in the dynamic simulations. The point-model will then be solved for each sub-section/pipe within each momentum control volume. The wall and interfacial friction from the point-model solution is then used instead of the regular friction terms in the dynamic simulation. Note that when the point-model predicts stratified flow, the result will be the same as for the regular friction terms described in section 2.10.1.

The wall shear stress integrated over a sub control volume can then be calculated from the unit-cell model as shown in equation (210) by using the wall shear stress in both the bubble-zone and in the slug.

$$\int_{A_k} \tau_{k,wall} dA = L\left(S_{k,bubble} \tau_{k,wall,UC,bubble} \left(1 - SF\right) + S_{k,slug} \tau_{k,wall,UC,slug} SF \right) \tag{209}$$

The interfacial shear stress integrated over a sub control volume to be used in the dynamic momentum equation is back-calculated from the steady state momentum balance, including the total wall friction from both the bubble-zone and the slug as shown in equation (211).

$$\int_{A_k} \tau_i dA = L\frac{\left[\left(\rho_l-\rho_g\right)g\sin\theta-\tau_{wg,bubble}\frac{S_{g,bubble}}{A_{g,bubble}}\left(1-SF\right)+\tau_{wl,bubble}\frac{S_{l,bubble}}{A_{l,bubble}}SF+\tau_{wl,slug}\frac{S_{l,slug}}{A_{l,slug}}\left(1-SF\right)\right]}{\left(\frac{1}{A_{l,bubble}}+\frac{1}{A_{g,bubble}}\right)}$$

(210)

Note that additional losses from for instance valves are not included in equation (211), but are instead included directly in the dynamic momentum equation. Both the wall and interfacial shear stress are still treated implicit when calculated from the unit-cell model. The resulting wall and interfacial shear stress are divided by the velocity and velocity difference between gas and liquid respectively, so that they can be applied to both the diagonal and right hand side in the momentum equation instead of only being an explicit term on the right hand side.

2.16 Steady state solver

2.16.1 Momentum

The unit-cell model presented in section 2.15 can be used to initialize a simulation with steady state holdup, velocities and pressure. The steady state solver currently only works with a closed inlet and open outlet, for which the unit-cell model is solved for each section in the pipeline, starting from the last cell. The unit-cell model requires superficial velocities as input, which can be found from the mass rate, density and cross-sectional area. At steady state the mass rate will be constant in each section in the pipeline, so that the mass rate in any section can be found by summing the mass rate in each section starting with the mass flow specified at the closed inlet border. Mass sources or mass sinks located in the pipeline are also included in the calculation. Since the superficial velocity (and fluid properties) is calculated from the density, but the density is a function of the pressure which is still unknown, solution of the steady state solver becomes and iterative procedure. The iteration procedure to solve for the steady state in a section is as follows:

List 4: Iterative procedure to solve pressure, velocity and holdup using the unit-cell model, starting at the last section.

1. Calculate the pressure from the pressure gradient (0 in the first iteration) and the pressure in the section to the right
2. Calculate fluid properties
3. Calculate superficial velocities from mass rate, density and area
4. Solve unit-cell model
5. Update average holdup and slug fraction weighted velocities, and store pressure gradient
6. Set masses from density and holdup
7. Repeat until change in pressure is below specified tolerance.

Since the steady state solver is solved non-staggered for the sections, while the momentum in the dynamic solver is solved staggered at the borders, the velocities at the borders are calculated from the section velocities after the steady state solver is finished. Note that the temperature is not a part of the steady state solver, and will be as specified from the initial conditions. It is also possible to force a stratified solution for the steady state solver.

The number of sections used by the steady state solver is set by using a grid coefficient calculated as the maximum of the specified minimum bubble coefficient and 100 times the mean pipe diameter in the pipeline. However, it is also ensured that no less than 16 sections are used. It is also ensured that no sections spans multiple pipes in the steady state solver; sections are split at all static borders. After the steady state solver is finished, sections might be merged or split again to obtain the grid one will get by using the regular grid coefficients. The user can however also specify not to modify the grid when calculating the steady state.

Additional losses due to valves or expansion/contraction losses are also added to the point model by K-values (see equation (207)). User-specified forces (see section Appendix C.4) does not affect the steady state momentum balance, so there is no point adding them to the unit-cell model. The pressure drop is however included in the steady state solver when calculating the pressure in one cell by the pressure drop and the pressure in the cell to the right.

2.16.2 Energy

The steady state solver can also calculate the steady state temperature profile in the pipe. At steady state conditions, and assuming flow from left to right with a first order upwind scheme, the energy equation becomes:

$$\begin{aligned}\dot{M}_{k,J}^{n+1}h_{k,j}^{n+1} &= \dot{M}_{k,J-1}^{n+1}h_{k,j-1}^{n+1} \\ &+\sum_{m}Q_{k,m} +\dot{M}_{k,j}^{src}\left(h_{k,j}^{src}+\frac{\left(u_{k,j}^{src}\right)^2}{2}\right)-M_{k,j}^{n}u_{k,j}^{n+1}g\frac{1}{L_j^n}\int_{L_j^n}\sin\left(\theta\left(x\right)\right)dx\end{aligned} \tag{211}$$

Here the mass fluxes in and out of the section has been simplified and replaced by the symbols $\dot{M}_{k,J-1}^{n+1}$ and $\dot{M}_{k,J}^{n+1}$ respectively. At steady state the mass flux in and out of a section are related as follows:

$$\dot{M}_{k,J} = \dot{M}_{k,J-1} + \dot{M}_{k,j}^{src} \tag{212}$$

Equation (212) is a linear equation for the steady state specific enthalpy in section j:

$$\begin{aligned}&\left(\dot{M}_{k,J}^{n+1} + \sum_m \frac{1}{R_{cond,km}^{n+1} c_k^{n+1}\left(p_k^{n+1}, T_k^{n+1}\right)}\right) h_{k,j}^{n+1} = \dot{M}_{k,J-1}^{n+1} h_{k,j-1}^{n+1} \\ &+\sum_m \frac{h_k\left(p_k^{n+1}, T_m^{n+1}\right) - 0}{R_{cond,km}^{n+1} c_k^{n+1}\left(p_k^{n+1}, T_k^{n+1}\right)} + \dot{M}_{k,j}^{src}\left(h_{k,j}^{src} + \frac{\left(u_{k,j}^{src}\right)^2}{2}\right) - M_{k,j}^{n} u_{k,j}^{n+1} g \frac{1}{L_j^n} \int_{L_j^n} \sin\left(\theta\left(x\right)\right) dx\end{aligned} \tag{213}$$

Here equation (114) has been inserted for the heat transfer terms, yielding coefficients multiplied by the steady state specific enthalpy. Note that in the dynamic energy equation we solve for the change in specific enthalpy, which results in an additional term with the specific enthalpy from the previous time step on the right hand side. This term is however not included in the right hand side in the steady state energy equation, as we solve directly for the specific enthalpy (term shown as a zero on the right hand side on purpose). In opposite to when solving for pressure, velocity and holdup where we start at the outlet and iterate towards the inlet, the steady state energy equation must be solved in the opposite direction. As the flow goes from left to right and the temperature and mass flow through the inlet border is constant, starting with the first section we can calculate the specific enthalpy $h_{k,j}^{n+1}$ from equation (214) by using the known specific enthalpy and mass flow in the left section.

The following iterative procedure is used:

List 5: Iterative procedure to solve the specific enthalpy/temperature in the pipeline, starting at the first section.

1. Update the energy equation for the all pipe walls
2. Calculate the fluid heat transfer terms for all sections
3. Solve pipe energy equation and update pipe wall specific enthalpy & temperature
4. For all section starting from the first one, solve equation (214) for the specific enthalpy using the specific enthalpy and mass flow in the left section
5. Repeat until convergence of the specific enthalpy for all sections and pipe walls

If specified to calculate the steady state condition, and the simulation is specified as non-isothermal, the steady state temperature field will be calculated in addition to pressure, velocity and holdup. Together this becomes a nested iterative procedure:

List 6: Iterative procedure to solve the steady state pressure, velocity, holdup and temperature.

1. Converge pressure, velocity and holdup from the point model for all sections as described in List 4.
2. Converge specific enthalpy/temperature for all sections and pipe wall as described in List 5.
3. Repeat until convergence

An example comparing the result from the steady state solver and the dynamic solution is shown in Figure 53. Note that the dynamic simulation starts from conditions given by the steady state solver. The test case is a horizontal pipe of length 10000 m with inner diameter 0.146 m and outlet pressure is 10 bar, using water and air mass rates of 20 kg/s and 0.2 kg/s respectively. Two wall layers are used (steel and PVC), with an ambient temperature of 350 Kelvin and an inlet temperature of 260 Kelvin. 128 sections have been used, both for the fluids and for the pipe walls. As can be seen in Figure 53, the dynamic result is quite similar to the result of the steady state solver. The temperature can be seen to be increasing from the inlet, as the fluids have more time to heat through the pipe walls.

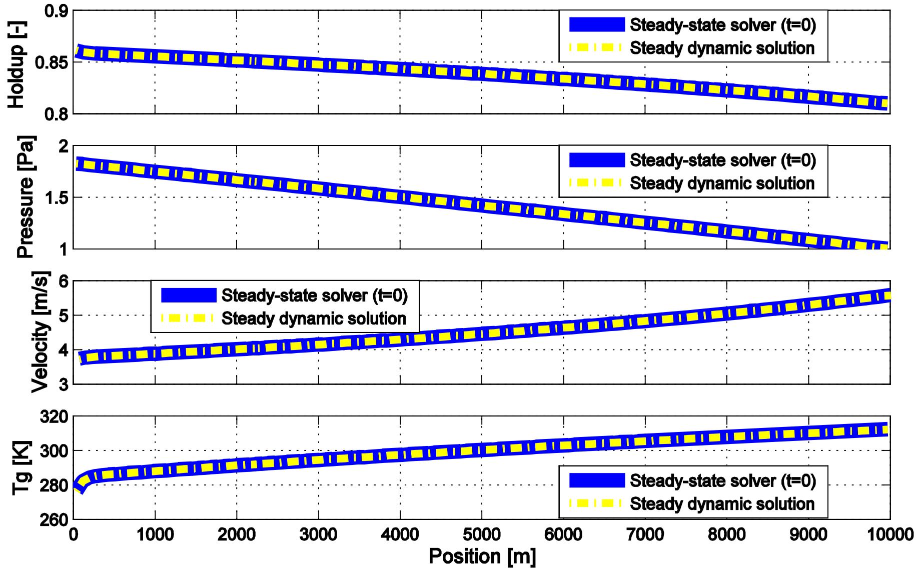


Figure 53: Steady state solution vs dynamic solution for a horizontal pipe test case.

2.17 Grid management

Since some of the control volumes are moving, sections can both grow and decrease in length. Because of this, certain grid management operations are needed: short sections are merged together with neighbouring sections of similar type, and long sections might be split into two shorter sections. Sections might also be converted from one type to another, for instance from bubble to slug, or slug to bubble.

2.17.1 Splitting

A section might be split into two smaller sections, if its length exceeds a user-specified length. There are in total four different user-specified coefficients that controls the splitting of slug and bubble sections. These are minimum and maximum coefficients for both bubble sections and for slugs. Bubble sections spanning static borders (i.e. bubble sections spanning multiple pipes) are tried to always be split at the static borders, unless any of the resulting section lengths after splitting is smaller than the specified minimum length. An example of this can be seen in Figure 54, where the small bubble section in the middle of the image is part of the larger bubble section to the left, since its length is shorter than the specified limit. Slug sections are not split at static borders. There is usually no reason to splitting the incompressible plugs, since the momentum balance is calculated over the entire plug unit anyways. The only reason for splitting incompressible plugs would be to get a better temperature profile over the incompressible plug unit. Splitting incompressible slug sections is thus not enabled by default. For compressible slugs on the other hand one can obtain a better pressure profile and distribution of gas bubbles in the slug by enabling the splitting.

A section will be merged with a neighbouring section of the same type if the length is smaller than the minimum grid coefficient, and split in two if the section length is larger than the maximum value of twice the minimum grid coefficient and the maximum grid coefficient.

Consequently, the length (Δx) of a section is bounded by the range shown in equation (215):

$$\Delta x_{\min} \leq \Delta x \leq \max\left(2 \cdot \Delta x_{\min}, \Delta x_{\max}\right) \tag{214}$$

Since it is possible to specify minimum and maximum coefficients at each different pipe, both the minimum and maximum coefficient is taken as the minimum value from all pipes present within the section.

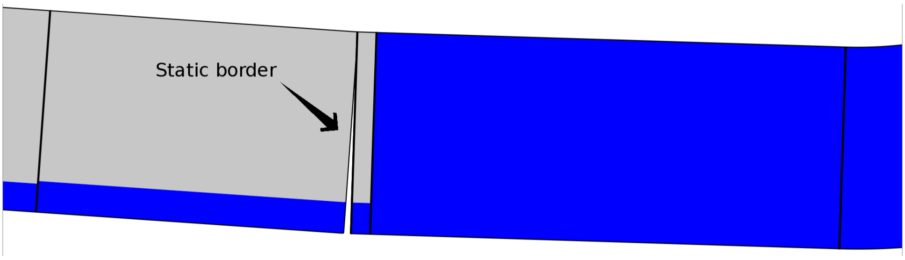


Figure 54: Example of a bubble section not being split at the static border in the middle of the image, since the length of the bubble section to the right of the static border would have become smaller than the minimum specified length. There is a difference in the pipe inclination to the left and right of the static border.

2.17.1.1 Avoiding single bubble sections between slugs

If a bubble section is located between two slugs, it will always be split in two. This is due to the fact that the momentum equation for the slugs is evaluated over the entire slug unit, and one would then end up not solving the momentum for single bubble sections between slugs. Single bubbles between a slug and a closed border will also always be split for the same reason. Since the slug border to the left and right of such a bubble-bubble border will typically move, one of the bubble sections would typically in the next time step becomes so small that it would have been merged with the neighbouring bubble, which would have been split at the middle again to avoid a single bubble section between slugs. This will work fine, but lead to unnecessary many merging and splitting operations. To avoid this it is possible to apply a velocity also to the bubble-bubble borders located between two slug borders. Ideally, this velocity should be so that the bubble-bubble border moves to the middle between the new positions of the slug borders. It will however complicate the equations to do this fully implicit, so the velocity of the bubble-bubble border is approximated using the current velocities of the neighbouring slug borders:

$$\begin{aligned} x_{J-1}^{n+1} &\approx x_{J-1}^{n} + \Delta t u_{b,J-1}^{n} \\ x_{J}^{n+1} &\approx x_{J}^{n} + \Delta t u_{b,J}^{n} \\ u_{b,j}^{n+1} &= \frac{\left(0.5\left(x_{J-1}^{n+1} + x_{J}^{n+1}\right) - x_{j}^{n}\right)}{\Delta t} \end{aligned} \tag{215}$$

This will make the bubble-bubble borders located between two moving slug borders move approximately to the middle between the new positions of the slug borders.

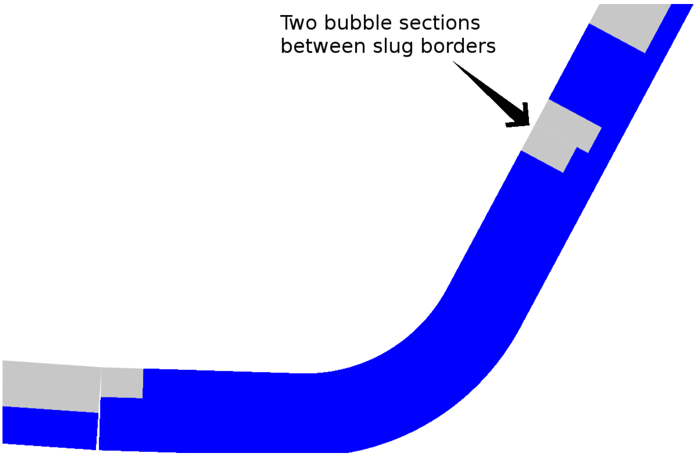


Figure 55: Illustration of two bubble sections moving between two slug sections.

2.17.2 Removal of sections

There are typically two ways a section can be removed as part of the grid management:

1. The length has become too small
2. The holdup has reached a specific limit

2.17.2.1 Length related removal

2.17.2.1.1 Bubble sections

If the length of a bubble section becomes smaller than the user-specified limit and if there is another bubble section on either side, they will be merged as long as the holdup in the neighbouring bubble sections is below the limit for conversion to slug. Short bubble sections between slugs and an open inlet or outlet will only be removed if they are travelling out of the pipeline, and if the length is shorter than a certain limit. If there is a slug section on both sides of the bubble section, there are several scenarios: If void in slug is enabled and the liquid holdup in the bubble section is above the user-specified limit, the bubble section will be converted to a slug section and the gas mass set as gas in liquid in the slug. If void in slug is not enabled, the gas mass will be removed and the bubble merged with one of the slugs.

2.17.2.1.2 Slug sections

Similar to the bubble sections a slug section will be merged with a neighbouring slug if the length is shorter than the minimum slug length coefficient. If there are no slugs on either side and the length becomes shorter than the user specified slug-kill limit, the slug section is first converted to a bubble section, which is then merged with one of the neighbouring bubble sections. The exception is when a slug is between a bubble section and an open inlet or outlet. Short slug sections between a bubble section and an open inlet or outlet will only be removed if they are travelling out of the pipeline, and if the length is shorter than a user-specified limit.

2.17.2.1.3 Merging

When merging two sections together, the middle border separating the two sections are removed and a new section created from the original two.
For bubble sections, the pressure in the new section is calculated as follows from the previous pressures and gas volumes:

$$p = \frac{p_1 V_{g,1} + p_2 V_{g,2}}{V_{g,1} + V_{g,2}} \tag{216}$$

For compressible slugs however the new pressure is calculated as shown in equation (217), but using the liquid volume.

2.17.2.2 Holdup related removal

If the liquid holdup in a bubble section has become larger than the user-specified limit, it will be converted to a slug section. If void in slug is enabled, the gas present in the bubble section will be put as gas bubbles in the liquid. Otherwise, the gas is distributed in the neighbouring sections, and the borders of the section are moved to make the section volume exactly equal to the mixture liquid volume (liquid holdup equals 1).

It might happen that the length of a bubble section is too small, and that the holdup has become larger than the user-specified limit simultaneously. In this case it is important to choose the correct way to remove the bubble section. If there is a slug on either side, or the length of the bubble is larger than the minimum specified slug length, the bubble should be converted to slug. If not, the bubble will be killed based on the length criteria, and merged with a neighbouring bubble section. There can also be several other situations where one will have to choose between different removal criteria, but these will not be described here.

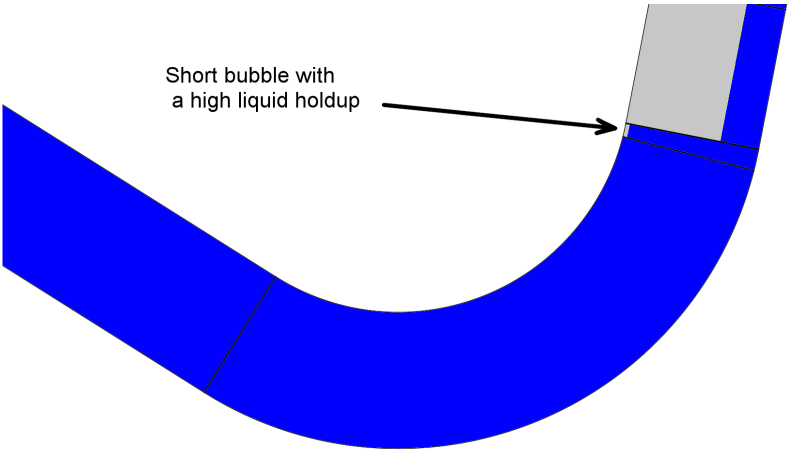


Figure 56: Illustration of a case when a bubble section has both become shorter than the minimum specified length, and has a liquid holdup higher than the specified limit for conversion to slug at the same time.

For slug sections on the other hand, the mixture liquid holdup will never decrease and always sum to 1 due to the way the slug borders are moved. If liquid is being drained from the slug, the slug will just become shorter in length. There is however also a limit for the maximum gas in liquid fraction in the slug (0.8). If the phase fraction of gas in liquid is larger than this limit, the slug will be converted to a bubble section.

2.17.2.3 Removal of pigs

Like slugs, a short pigs between a bubble section and an open inlet or outlet will only be removed if they are travelling out of the pipeline, and if the length is shorter than what is specified by a certain limit. Note that the pig typically has a constant mass during the simulation, but it is allowed to change mass when in contact with an open inlet or outlet to make it move out of the pipeline. If a pig has started to move out of the pipeline and then turns to move into the pipeline again, it will also be removed.

2.17.3 Inlet or outlet related removal

For the sections in contact with an open inlet or outlet, it is checked if a new bubble section or slug section should be inserted between the section itself and the inlet/outlet border. This might happen for instance if a slug that was previously moving out of the outlet has turned, and the holdup in the outlet section is smaller than the limit for conversion to slug. A bubble section will then be inserted between the slug and the outlet. Similarly one might have to insert a slug section between a bubble section and the outlet, if the holdup is the outlet section is larger than the conversion limit to slug.

2.17.4 Stopping slugs

When there are diameter changes in the pipeline, it might be unphysical for a slug to survive travelling past a large increase in pipe diameter. A typical example of this can be a slug entering a separator, which a slug would never survive in real life. This is for instance used in the simulations in Paper 1. It is possible for the user to specify certain locations (static border locations) in the pipeline where slugs should be killed. Slug borders that move past this location will then be stopped at the static border. The slug border will then have a zero velocity, so that there will be a net flux out of the slug front. The slug will then typically become smaller and smaller until it is so short that it is removed. If the velocity at the slug border that is stopped at the static border changes direction, the slug border is allowed to move as normal again.

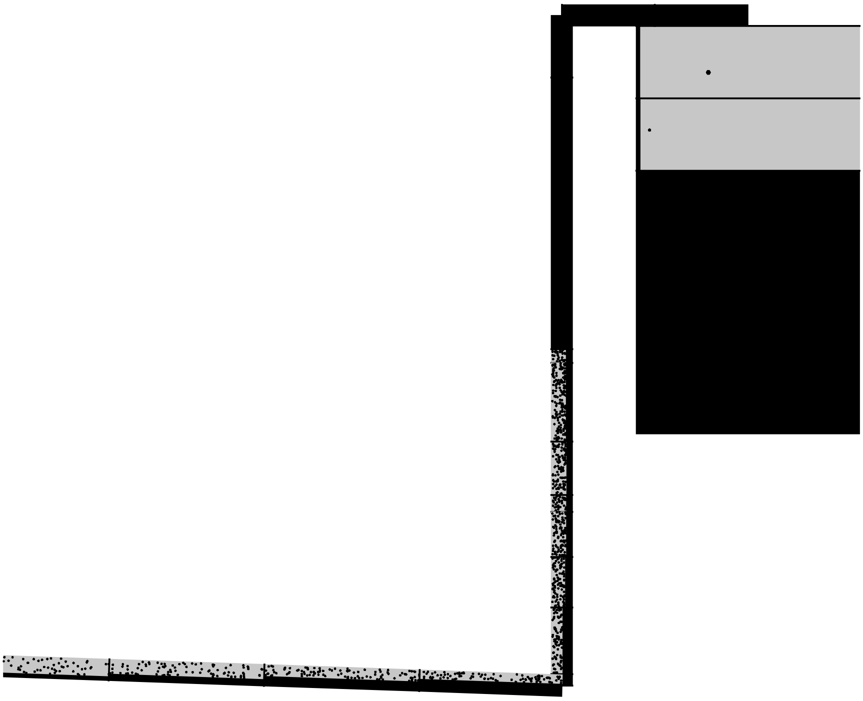


Figure 57: Image of a slug entering a separator. The right end of the slug has been stopped at the entrance to the separator.

2.17.5 Conservation of momentum and energy

When splitting, merging or converting sections, case is taken to ensure that momentum and energy is conserved.

For instance, the specific enthalpy in the new section when merging two sections is calculated as shown in equation (218) for each field (numbers 1 and 2 here referring to the two sections to be merged):

$$h_k = \frac{M_{k,1}h_{k,1} + M_{k,2}h_{k,2}}{M_1 + M_2} \tag{217}$$

The new temperature for each phase is then calculated from the new specific enthalpy. When splitting a section in two parts, the specific enthalpy remains unchanged (since specific enthalpy is energy per mass).

Regarding the conservation of momentum, what is important is that the momentum is updated in the momentum control volumes where the momentum is solved. Conserving the momentum for the two sections that are merged and updating the velocity in the new section would not have any effect, as momentum is not used/solved for the sections. One simple example is when only bubble sections exist: When merging two bubble sections a bubble-bubble border where momentum is solved will be removed, and this momentum is then distributed to the neighbouring bubble-bubble borders. The total momentum in the momentum control volumes and total mass times specific enthalpy in the sections for each mixture field is calculated and stored both before and after the grid management, so one can check that these properties are conserved during the grid management. Figure 58 shows the difference in total momentum and total mass times specific enthalpy for a selected simulation from Paper 1. It can be seen that both momentum and energy is conserved within a quite small numerical error though many different types of grid management operations happen during this simulation. Note that in opposite to the simulations in Paper 1, this simulation was run with the energy equations enabled with an adiabatic pipe wall.

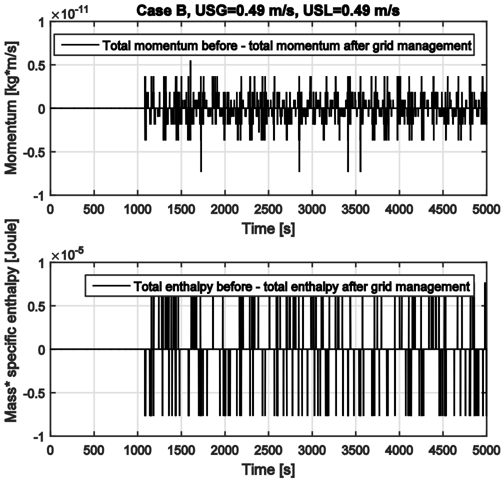


Figure 58: Conservation of momentum and energy during grid management for a selected simulation from Paper 1.

2.18 Fluid properties

Several different fluid properties are needed in the simulations for the gas, oil and water. These properties can be specified either as constant user defined values, assuming incompressible liquid, or by specifying the path to PVT-files.

2.18.1 Constant fluid properties

The default in Sluggit is to use constant fluid properties, except for the gas density. Assuming ideal gas law, the gas density can calculated from equation (219).

$$\rho_g = \frac{pM_{gas}}{RT} \tag{218}$$

Here, M_{gas} is a user-specified constant, and R is the ideal gas constant, which equals $8.3143 \frac{J}{mol \cdot K}$. Alternatively, it is possible to specify analytical functions for the fluid densities and viscosities in the input file. These functions are specified as text, and will be evaluated using the script functionality in the QtScript module in the Qt framework [83]. The function must be a text string, and the variables p and T are available in the script, representing the pressure (in Pa) and temperature (in Kelvin).

2.18.2 PVT-files

The fluid properties in Sluggit can also be specified by using PVT-files, for instance generated from PVT-programs like PVTSim Nova [84]. Note that to simulate incompressible slugs, the liquid must be treated as incompressible. The oil and water densities will thus still be taken from the global constant variables, even if a PVT-file with compressible densities is specified.

3 References

1. Bendiksen, K.H., *An experimental investigation of the motion of long bubbles in inclined tubes.* International Journal of Multiphase Flow, 1984. **10**(4): p. 467-483.
2. Benjamin, T.B., *Gravity currents and related phenomena.* Journal of Fluid Mechanics, 1968. **31**(02): p. 209-248.
3. Dumitrescu, D., *Strömung an Einer Luftblase im Senkrechten rohr.* Zeitschrift für Angewandte Mathematik und Mechanik, 1943. **23**(3): p. 139-149.
4. Jeyachandra, B.C., et al., *Drift-Velocity Closure Relationships for Slug Two-Phase High-Viscosity Oil Flow in Pipes.* SPE Journal, 2012. **17**(2): p. 593-601.
5. Joseph, D.D., *Rise velocity of a spherical cap bubble.* Journal of Fluid Mechanics, 2003. **488**: p. 213-223.
6. Viana, F., et al., *Universal correlation for the rise velocity of long gas bubbles in round pipes.* Journal of Fluid Mechanics, 2003. **494**: p. 379-398.
7. Stewart, H.B. and B. Wendroff, *Two-phase flow: Models and methods.* Journal of Computational Physics, 1984. **56**(3): p. 46.
8. Ishii, M. and T. Hibiki, *Thermo-Fluid Dynamics of Two-Phase Flow*. 2010: Springer New York.
9. Kjølaas, J., A. De Leebeeck, and S.T. Johansen, *Simulation of Hydrodynamic Slug Flow Using the LedaFlow Slug Capturing model*. 2013, BHR Group.
10. Kongsberg Oil and Gas Technologies. *LedaFlow*. 2015; Available from: http://www.kongsberg.com/en/kogt/products%20and%20services/flow%20assurance/.
11. SINTEF. *LedaFlow*. 2015; Available from: http://www.sintef.no/sintef-materialer-og-kjemi/programvare-2/ledaflow/.
12. Kjølaas, J., *Plug propagation in multiphase pipelines: Modeling and small scale experiments*, in *Faculty of Energy and Process Engineering*. 2007, NTNU: Trondheim.
13. Kjeldby, T., *Lagrangian Three-phase Slug Tracking Methods*, in *Department of Energy and Process Engineering*. 2013, NTNU: Trondheim.
14. Dukler, A.E. and M.G. Hubbard, *A Model for Gas-Liquid Slug Flow in Horizontal and Near Horizontal Tubes.* Industrial & Engineering Chemistry Fundamentals, 1975. **14**(4): p. 337-347.
15. Schlumberger. *OLGA*. 2016; Available from: https://www.software.slb.com/products/olga.
16. Issa, R.I. and M.H.W. Kempf, *Simulation of slug flow in horizontal and nearly horizontal pipes with the two-fluid model.* International Journal of Multiphase Flow, 2003. **29**(1): p. 69-95.
17. Courant, R., K. Friedrichs, and H. Lewy, *Über die partiellen Differenzengleichungen der mathematischen Physik.* Mathematische Annalen, 1928. **100**(1): p. 32-74.
18. Morin, A., T. Flåtten, and S.T. Munkejord, *A Roe scheme for a compressible six-equation two-fluid model.* International Journal for Numerical Methods in Fluids, 2013. **72**(4): p. 478-504.
19. Courant, R., E. Isaacson, and M. Rees, *On the solution of nonlinear hyperbolic differential equations by finite differences.* Communications on Pure and Applied Mathematics, 1952. **5**(3): p. 243-255.

20. Pascau, A., C. Pérez, and D. Sánchez, *A well-behaved scheme to model strong convection in a general transport equation.* International Journal of Numerical Methods for Heat & Fluid Flow, 1995. **5**(1): p. 13.
21. Leonard, B.P., *Simple high-accuracy resolution program for convective modelling of discontinuities.* International Journal for Numerical Methods in Fluids, 1988. **8**(10): p. 1291-1318.
22. Gaskell, P.H. and A.K.C. Lau, *Curvature-compensated convective transport: SMART, A new boundedness- preserving transport algorithm.* International Journal for Numerical Methods in Fluids, 1988. **8**(6): p. 617-641.
23. Sweby, P.K., *High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws.* SIAM Journal on Numerical Analysis, 1984. **21**(5): p. 995-1011.
24. Harten, A., *High resolution schemes for hyperbolic conservation laws.* Journal of Computational Physics, 1983. **49**(3): p. 357-393.
25. Wikipedia. *Flux Limiter*. 2016; Available from: http://en.wikipedia.org/wiki/Flux_limiter.
26. Segal, G., et al., *User Manual of the Delft incompressible flow solver*. 2000.
27. Hassanzadeh, H., J. Abedi, and M. Pooladi-Darvish, *A comparative study of flux-limiting methods for numerical simulation of gas–solid reactions with Arrhenius type reaction kinetics.* Computers & Chemical Engineering, 2009. **33**(1): p. 133-143.
28. *Approximation Schemes for convective term*. 2016; Available from: http://www.cfd-online.com/Wiki/Approximation_Schemes_for_convective_term.
29. Varonos, A. and G. Bergeles, *Development and assessment of a variable-order non-oscillatory scheme for convection term discretization.* International Journal for Numerical Methods in Fluids, 1998. **26**(1): p. 1-16.
30. Meese, E.A., *Finite Volume Methods for the Incompressible Navier-Stokes Equations on Unstructured Grids*, in *Department of Applied Mechanics, Thermo and Fluid Dynamics*. 1998, NTNU.
31. Ransom, V.H., *Numerical Benchmark Test No. 2.1: Faucet Flow.* 1987. **3**(1-4): p. 465-467.
32. Vatsa, V., M. Carpenter, and D. Lockard, *Re-evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications*, in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. 2010, American Institute of Aeronautics and Astronautics.
33. Prinja, G.K., *Adaptive Solvers for Elliptic and Parabolic Partial Differential Equations*. 2010, The University of Manchester.
34. Denner, A.K., *Experiments on temporal variable step BDF2 algorithms*. 2014, University of Wisconsin-Milwaukee.
35. Eckert, S., et al., *A BDF2 integration method with step size control for elasto-plasticity.* Computational Mechanics, 2004. **34**(5): p. 377-386.
36. Hairer, E., S.P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I.* Vol. 8. 1993.
37. Emmrich, E., *Stability and error of the variable two-step BDF for semilinear parabolic problems.* Journal of Applied Mathematics and Computing, 2005. **19**(1): p. 33-55.
38. Biberg, D., *An explicit approximation for the wetted angle in two-Phase stratified pipe flow.* The Canadian Journal of Chemical Engineering, 1999. **77**(6): p. 1221-1224.
39. Johansen, S., T. Ytrehus, and K. Einarsrud, *Modeling of multiphase flows*. 2011: Norwegian University of Science and Technology, course material.

40. Thon, B.O., *Friction Models for Oscillating Flow in a U-tube*, in *Department of Physics*. 2014, NTNU.
41. White, F.M., *Fluid mechanics*. 2011, New York, N.Y.: McGraw Hill.
42. Incropera, F.P., et al., *Fundamentals of Heat and Mass Transfer*. 2006: John Wiley & Sons.
43. Gnielinski, V., *New Equations for Heat and Mass Transfer in Turbulent Pipe and Channel Flow.* International Journal of Chemical Engineering, 1976. **16**: p. 10.
44. *LedaFlow user manual, version 2.0.250.021*. 2015.
45. Churchill, S.W. and M. Bernstein, *A Correlating Equation for Forced Convection From Gases and Liquids to a Circular Cylinder in Crossflow.* Journal of Heat Transfer, 1977. **99**(2): p. 300-306.
46. Isaacson, E. and H.B. Keller, *Anslysis of Numerical Methods*. 1994: Dover Publications.
47. Press, W.H., et al., *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 2007: Cambridge University Press. 1256.
48. Moody, L.F., *Friction factors for pipe flow.* Transactions of the ASME, 1944. **66**(8): p. 671–684.
49. Haaland, S.E., *Simple and Explicit Formulas for the Friction Factor in Turbulent Pipe Flow.* Journal of Fluids Engineering, 1983. **105**(1): p. 89-90.
50. Andritsos, N. and T.J. Hanratty, *Influence of Interfacial Waves in Stratified Gas-Liquid Flows.* AIChE J., 1987. **33**(3): p. 444-454.
51. Moeck, E.O. and J.W. Stachiewicz, *A droplet interchange model for annular-dispersed, two-phase flow.* International Journal of Heat and Mass Transfer, 1972. **15**(4): p. 637-653.
52. Cordell, J.L., *Conventional pigs - what to use and why*, in *Pipeline pigging and inspection technology*. 1992: Houston, Texas.
53. Govan, A.H., *Modelling of vertical annular and dispersed two-phase flows* 1990, Imperial College London.
54. Clift, R., J.R. Grace, and M.E. Weber, *Bubbles, Drops and Particles*. 1978: New York: Academic Press.
55. U.S. Nuclear Regulatory Commission, *TRAC-M/FORTRAN 90 Version 3.0 Theory Manual*. 2001, U.S. Nuclear Regulatory Commission.
56. Nydal, O.J. and P. Andreussi, *Gas Entrainment In Liquid Slugs*. International Society of Offshore and Polar Engineers.
57. Brauner, N. and A. Ullmann, *Modelling of gas entrainment from Taylor bubbles. Part A: Slug flow.* International Journal of Multiphase Flow, 2004. **30**(3): p. 239-272.
58. Skartlien, R., et al., *A gas entrainment model for hydraulic jumps in near horizontal pipes.* International Journal of Multiphase Flow, 2012. **43**: p. 39-55.
59. Manolis, I.G., *High Pressure Gas–Liquid Slug Flow*. 1995, Imperical College, London, UK.
60. Chanson, H., *Air Bubble Entrainment in Free-Surface Turbulent Shear Flows*, in *Air Bubble Entrainment in Free-Surface Turbulent Shear Flows*. 1996, Academic Press: San Diego.
61. Bonizzi, M. and R.I. Issa, *A model for simulating gas bubble entrainment in two-phase horizontal slug flow.* International Journal of Multiphase Flow, 2003. **29**(11): p. 1685-1717.
62. Malnes, D., *Slip relations and momentum equations in two-phase flow*. 1979, IFE, Report EST-1.

63. Issa, R.I., *Simulation of intermittent flow in multiphase oil and gas pipelines*, in *Seventh International Conference on CFD in the Minerals and Process Industries*. 2009: Melbourne, Australia.
64. Danielson, T.J., et al., *Simulation of Slug Flow in Oil and Gas PIpelines Using a New Transient Simulator*. Offshore Technology Conference.
65. Kjølaas, J. and S.T. Johansen, *Terrain Slugging in a High-Pressure Pipeline-Riser System - Large Scale Experiments and Predictions With LedaFlow*, in *9th North American Conference on Multiphase Technology*. 2014, BHR Group: Banff, Canada.
66. Barnea, D. and Y. Taitel, *Interfacial and structural stability of separated flow.* International Journal of Multiphase Flow, 1994. **20**: p. 387-414.
67. Sanderse, B., M. Haspels, and R.A.W.M. Henkes, *Simulation of Elongated Bubbles in a Channel Using the Two-Fluid Model.* Journal of Dispersion Science and Technology, 2015. **36**(10): p. 1407-1418.
68. Fullmer, W.D., V.H. Ransom, and M.A. Lopez de Bertodano, *Linear and nonlinear analysis of an unstable, but well-posed, one-dimensional two-fluid model for two-phase flow based on the inviscid Kelvin–Helmholtz instability.* Nuclear Engineering and Design, 2014. **268**: p. 173-184.
69. Munkejord, S.T., *Comparison of Roe-type methods for solving the two-fluid model with and without pressure relaxation.* Computers & Fluids, 2007. **36**(6): p. 1061-1080.
70. Bendiksen, K. and M. Espedal, *Onset of slugging in horizontal gas-liquid pipe flow.* International Journal of Multiphase Flow, 1992. **18**(2): p. 237-247.
71. Lawrence, C., et al., *Mechanistic Models for Three-Phase Stratified and Slug Flows with Dispersions*. BHR Group.
72. Lawrence, C., et al., *Validation of a Physically-Based Model for Slug Initiation and Evolution in Hydrodynamic Slug Flow*. BHR Group.
73. Johansen, M. and O.J. Nydal, *Experiments on large gas bubbles in multiphase gas-oil-water pipe flow*, in *HEFAT 2005, 4th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamic*. 2005.
74. Nydal, O.J., *Experiments in downwards flow on stability of slug fronts*, in *Third International Conference on Multiphase Flow*. 1998: Lyon, France.
75. Nuland, S., *Bubble front velocity in horizontal slug flow with viscous Newtonian-, shear thinning- and Bingham fluids*, in *Third International Conference on Multiphase Flow*. 1998: Lyon, France.
76. Chen, C., *Unified Theory on Power Laws for Flow Resistance.* Journal of Hydraulic Engineering, 1991. **117**(3): p. 371-389.
77. Wallis, G.B., *One dimensional two phase flow*. 1969, New York: McGraw-Hill.
78. Bendiksen, K., *An experimental investigation of the motion of long bubbles in inclined tubes.* International Journal of Multiphase Flow, 1984. **10**(4): p. 17.
79. Moissis, R. and P. Griffith, *Entrance effects in a developing slug flow.* Journal of Heat Transfer, 1962. **84**: p. 29-39.
80. Bendiksen, K.H., et al., *The Dynamic Two-Fluid Model OLGA: Theory and Application* SPE Production Engineering, 1991. **6**(2): p. 10.
81. Eskerud Smith, I., J. Nossen, and T.E. Unander, *Improved Holdup and Pressure Drop Predictions for Multiphase Flow with Gas and High Viscosity Oil*. 2013, BHR Group.
82. Gregory, G.A., M.K. Nicholson, and K. Aziz, *Correlation of the Liquid Volume Fraction in The Slug for Horizontal Gas-Liquid Slug Flow.* International Journal of Multiphase Flow, 1978. **4**: p. 33-39.
83. Qt Company. *Qt Project*. 2016; Available from: http://www.qt.io/.

84. Calsep, *PVTSim Nova*. 2014.

85. Whitaker, S., *Advances in Theory of Fluid Motion in Porous Media.* Industrial & Engineering Chemistry, 1969. **61**(12): p. 14-28.

86. Bratland, O., *Pipe Flow 1: Single-phase Flow Assurance*. 2009.

87. Rahimi, S. *Contraction, Expansion, Pressure Drop*. 2011; Available from: http://www.chemwork.org/PDF/board/Expansion,%20Contraction,%20Pressure%20Drop.pdf.

88. Wikipedia. *Automatic differentiation*. 2016; Available from: http://en.wikipedia.org/wiki/Automatic_differentiation.

Appendix A Useful relations

The starting point for derivation of conservation equations in fluid mechanics is the Reynolds transport theorem. The transport of an arbitrary property ϕ through a control volume $V*$ with surface $A*$ is ([41]) :

$$\frac{d}{dt}\int_{V*}\phi dV = \int_{V*}\frac{\partial\phi}{\partial t}dV + \int_{A*}\phi\mathbf{u}_b\cdot\mathbf{n}dA \tag{219}$$

The outward pointing normal vector of $A*$ is denoted **n**, while the surface of the volume travels at velocity $\mathbf{u}_b$. Another fundamental relation is the Gauss divergence theorem:

$$\int_V(\nabla\cdot\mathbf{F})dV = \int_A(\mathbf{F}\cdot\mathbf{n})dA \tag{220}$$

By dividing the Reynolds transport theorem for phase *k* by the volume and assuming that *V* is fixed, we get:

$$\frac{1}{V}\frac{d}{dt}\int_{V_k}\phi_k dV = \frac{1}{V}\int_{V_k}\frac{\partial\phi_k}{\partial t}dV + \frac{1}{V}\int_{A_k}\phi_k\mathbf{u}_b\cdot\mathbf{n}_k dA \tag{221}$$

We now introduce the concept of a *volume averaged* quantity (sometimes also called the extensive average, $\langle\phi_k\rangle$) and *intrinsic averaged* quantity ($\langle\phi_k\rangle^i$) ([39]):

$$\langle\phi_k\rangle = \frac{1}{V}\int_{V_k}\phi_k dV = \frac{V_k}{V}\frac{1}{V_k}\int_{V_k}\phi_k dV = \alpha_k\langle\phi_k\rangle^i \tag{222}$$

Here we have also used the definition of the volume phase fraction:

$$\alpha_k = \frac{V_k}{V} \tag{223}$$

Re-arranging, we can now write equation (222) as:

$$\left\langle \frac{\partial \phi_k}{\partial t} \right\rangle = \frac{\partial \left\langle \phi_k \right\rangle}{\partial t} - \frac{1}{V} \int_{A_k} \phi_k \mathbf{u}_b \cdot \mathbf{n}_k dA \tag{224}$$

Equation (225) is useful when volume averaging transient terms. A similar expression for spatial derivatives can be obtained by use of Slattery's theorem ([85]). The complete derivation will not be shown here, but the result is ([39]):

$$\left\langle \nabla \phi_k \right\rangle = \nabla \left\langle \phi_k \right\rangle + \frac{1}{V} \int_{A_k} \phi_k \mathbf{u}_b \cdot \mathbf{n}_k dA \tag{225}$$

Appendix B Demonstration simulations

This section presents simulations that demonstrates different capabilities of the Sluggit code. For instance, section Appendix B.2 compares the difference from doing simulations with incompressible and compressible slugs.

Appendix B.1 Single phase pressure drop

This case study shows the pressure drop in single phase water and single phase gas flow. The fluid enters through a mass flow boundary at the inlet, while a pressure boundary with a constant pressure of 101300 Pa is used at the outlet. For the gas, the reverse case is also shown just to illustrate that this also works and gives the same result (mass flow boundary at outlet and pressure boundary at inlet). For the single phase gas the mass flow is 0.0238 kg/s, which gives a velocity of 10 m/s (the pipe diameter is 0.05 m). For the single phase water case, the mass flow and corresponding velocity are 1.9618 kg/s and 1 m/s respectively. Note that in the water case, the holdup limit for conversion to slug was set to a large number larger than one to never convert the bubble section to a slug object. The theoretical and simulated pressure gradient is shown in the title in Figure 59 and Figure 60, and can be seen to match each other exactly. This is of course not a surprising result, but is nevertheless presented here to prove that the basic physics in the code is correct. Figure 60 also shows that the case with reversed flow direction gives the same result as the normal case.

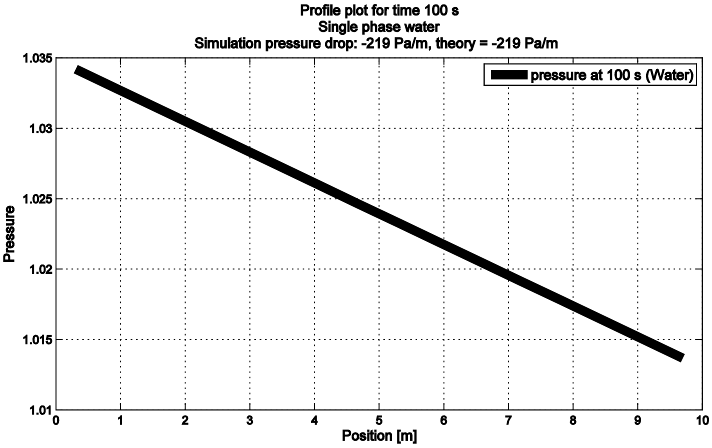


Figure 59: Pressure drop from single phase water flow, using a pressure boundary of 101300 Pa at the outlet.

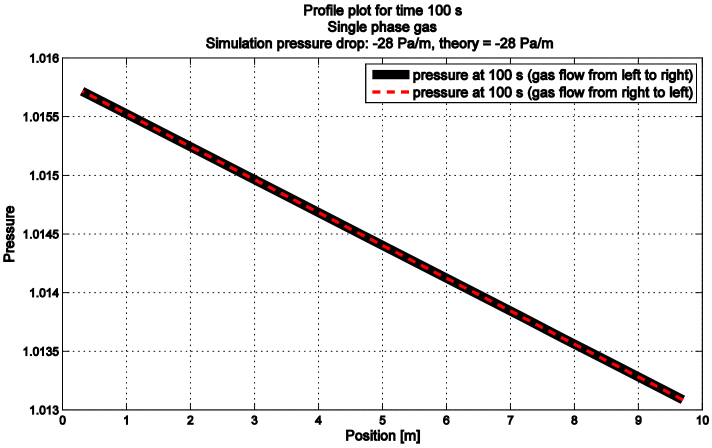


Figure 60: Pressure drop from single phase gas flow, showing results for both flow from left to right and flow from right to left. The simulation with flow from right to left is plotted reversed.

Appendix B.2 Symmetry test number 1

This case study presents the results from simulations of liquid flowing from left to right and right to left, to check that the code handles this symmetrically. Two test cases are presented here, one with an open inlet and outlet, and one with closed inlet and outlet. The pipe has an inner diameter of 0.2 m and is inclined by 10 degrees (see visualization in Figure 61, Figure 62, Figure 64 and Figure 65). Figure 63 and Figure 66 shows the holdup in the two cases, compared to the holdup from the simulation with reversed flow direction. It can be seen that the difference in the holdup when comparing against a simulation performed in the opposite direction is quite small, indicating that the code handles the flow symmetrically. Note that the weighting between the stratified and annular friction models are handled symmetrically in these simulations (see discussion below equation (80)).

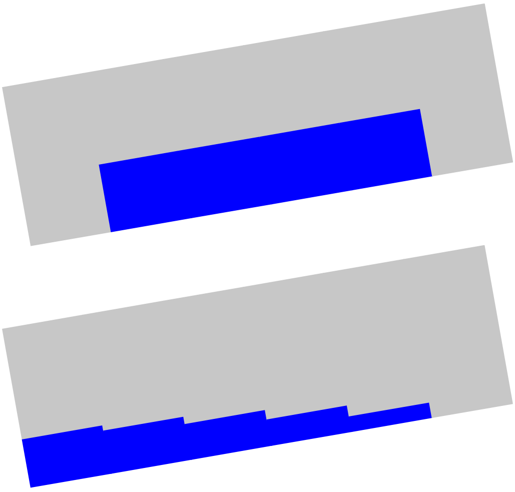


Figure 61: Visualization of the holdup in a pipe with liquid sliding left at times 0 seconds and 2 seconds. The pipe is open in both ends.

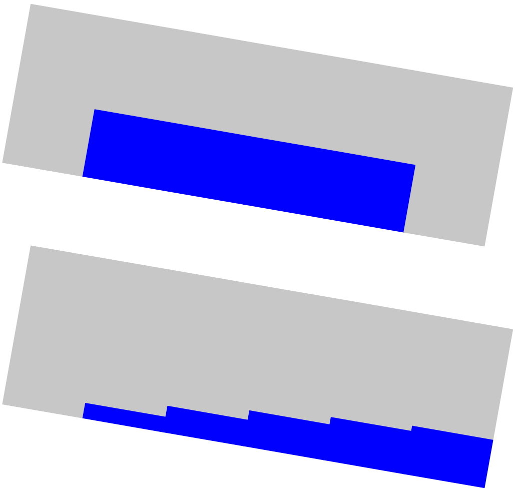


Figure 62: Visualization of the holdup in a pipe with liquid sliding right at times 0 seconds and 2 seconds. The pipe is open in both ends.

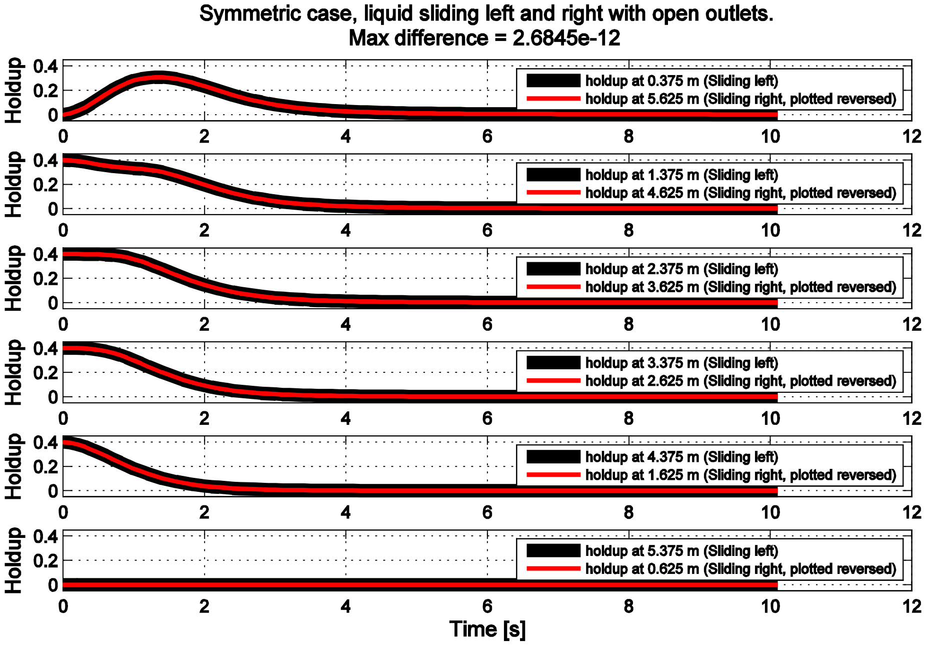


Figure 63: Liquid holdup plotted against time for different positons in the pipe, for both flow from left to right and right to left in a pipe with open inlet and open outlet.

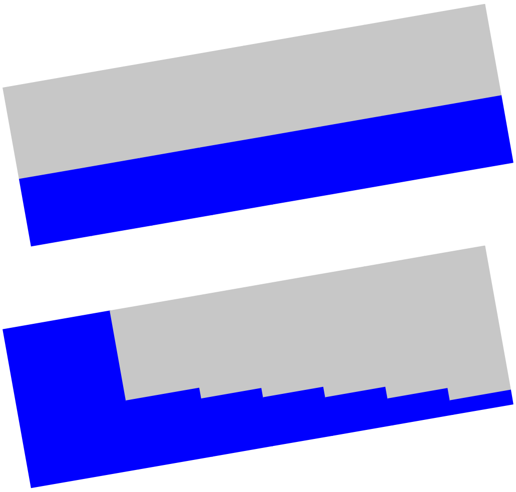


Figure 64: Visualization of the holdup in a pipe with liquid sliding left at times 0 seconds and 1.6 seconds. The pipe is closed in both ends.

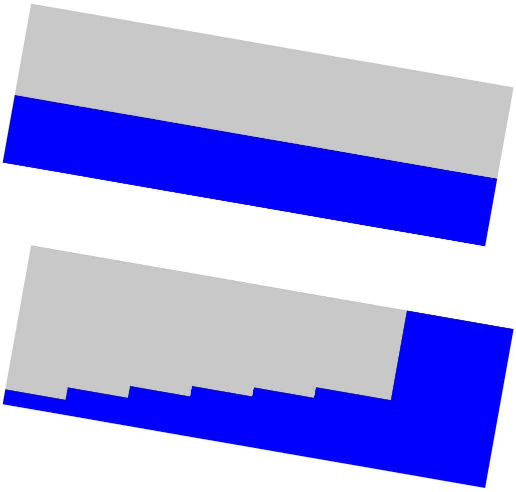


Figure 65: Visualization of the holdup in a pipe with liquid sliding right at times 0 seconds and 1.6 seconds. The pipe is closed in both ends.

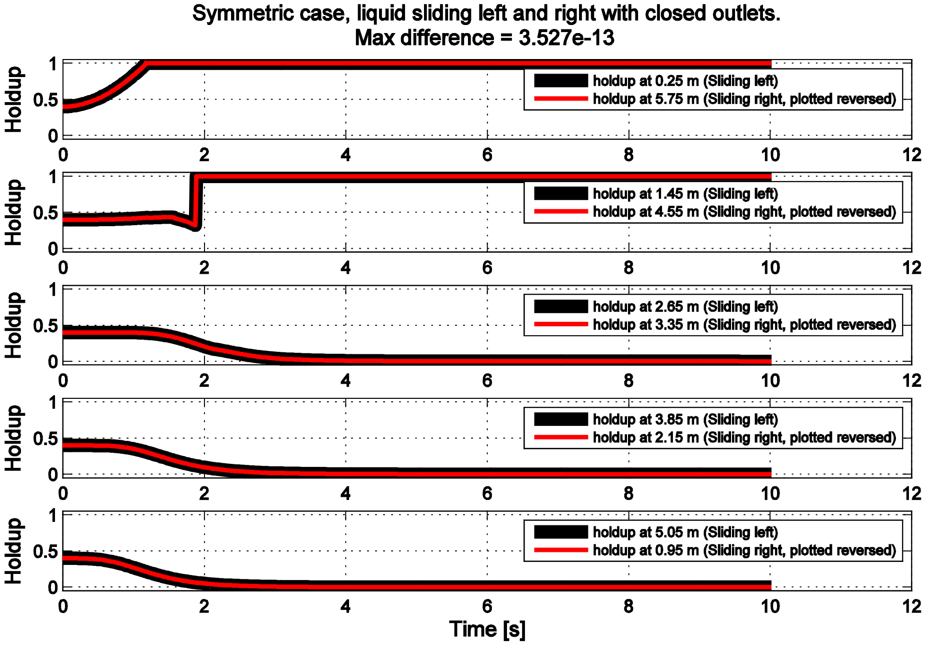


Figure 66: Liquid holdup plotted against time for different positons in the pipe, for both flow from left to right and right to left in a pipe with closed inlet and closed outlet.

Appendix B.3 Simulations with gas-oil compared to simulations with gas-water

This case study presents simulations of two-phase gas-water flow compared to gas-oil flow when using the same fluid properties for oil and water (meaning that the results should be identical). The simulations were performed with the energy equations and wall-heat transfer enabled, droplet entrainment, and with gas in the liquid in the bubble sections and in the slugs enabled. The mass rates used for gas and liquid were 0.0021 and 0.295 kg/s respectively. The simulations are performed using a s-shaped riser geometry, as shown in Figure 67. Figure 68 shows a comparison between the gas and liquid temperature in the oil and water case at two positions in the pipe. It can be seen that the difference in results are zero, which indicates an identical result. The oil and water field are in other words treated similarly, as expected.

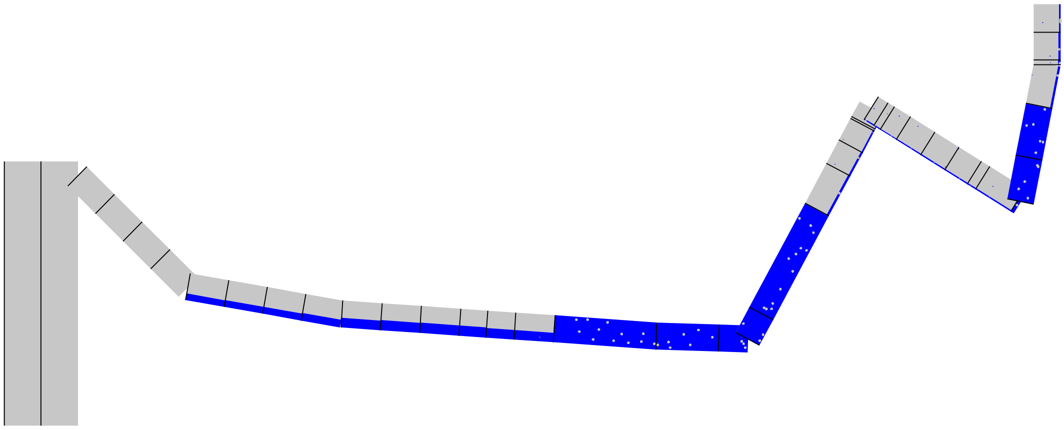


Figure 67: Geometry of s-shaped riser test case.

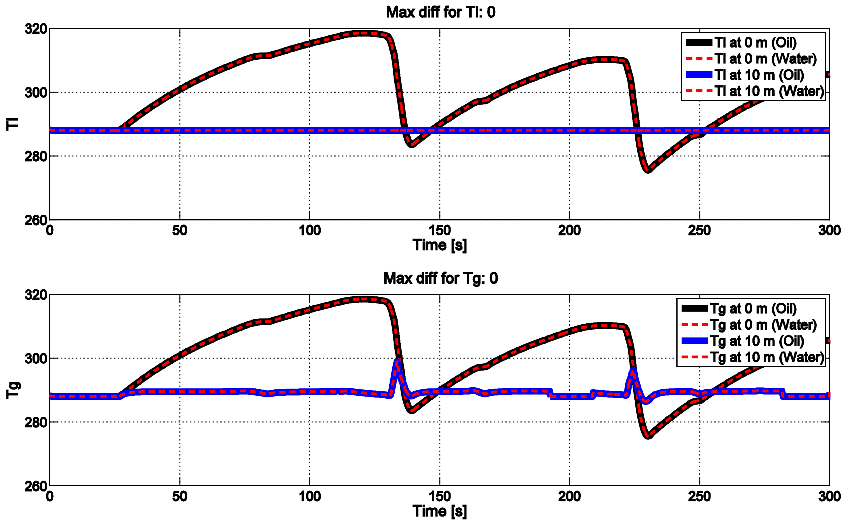


Figure 68: Comparison of the gas and liquid temperature for the oil and water simulations in the s-shaped riser.

Appendix B.4 Incompressible vs compressible slugs

This study presents the results from simulations of a severe slugging experiment from Paper 1, showing the difference in slugging period using incompressible slugs, compressible slugs, compressible slugs with gas in liquid, and compressible slugs with gas in liquid and the energy equations enabled. The simulation with the energy equations enabled were run without any heat transfer with the pipe walls. For the simulations with compressible liquid, the liquid density was modelled by the following equation: $680 - 6.75 \cdot p \cdot 10^{-7}$, with p being the pressure in Pa. The results from the different simulations are compared to the experimental data in Figure 69.

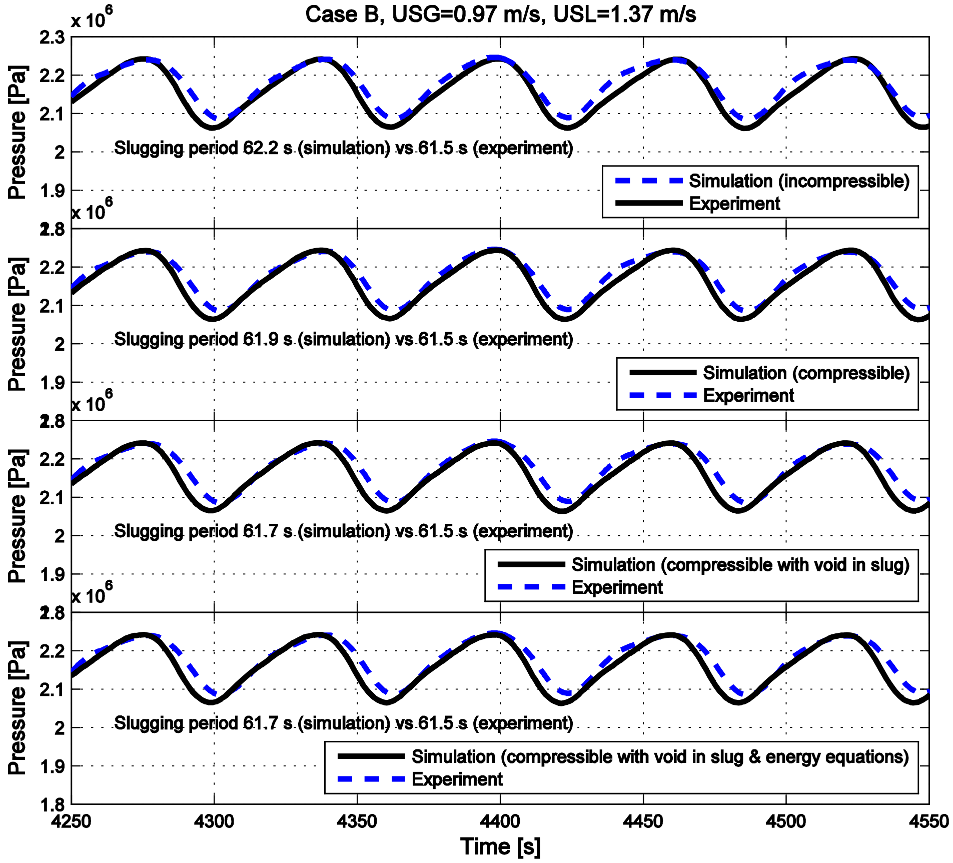


Figure 69: Experimental vs simulated pressure in the start of the pipeline for a selected experiment from Paper 1 with USG=0.97 m/s and USL=1.37 m/s.

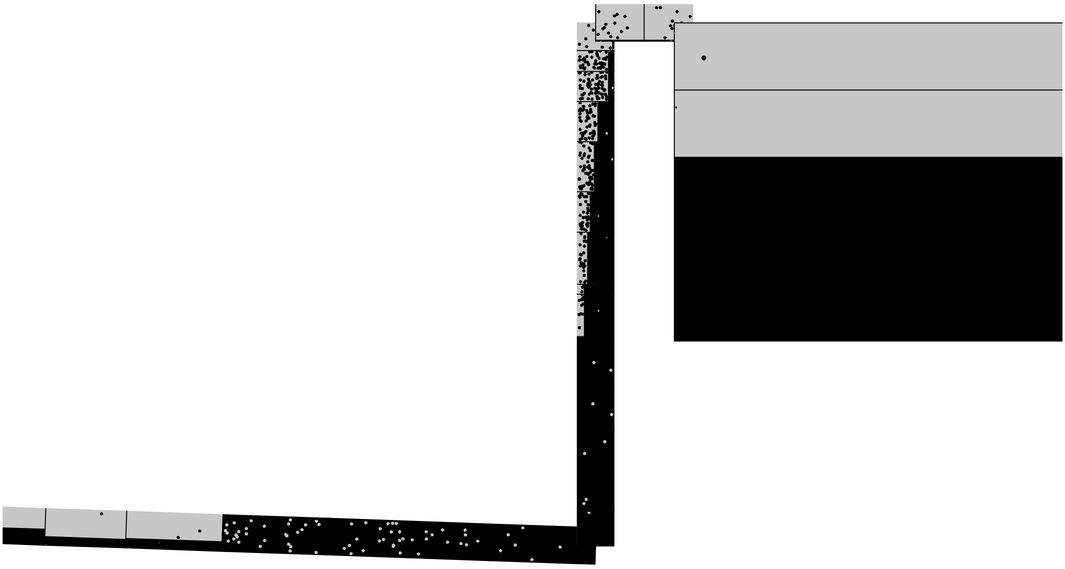


Figure 70: Figure showing the riser and separator in a simulation from Paper 1, with compressible slugs, gas in liquid and the energy equations enabled.

From Figure 69 it can be seen that there is not much difference in the results, though this could probably vary depending in the chosen experiment. The temperature was found to fluctuate most at the star of the pipeline due to the pressure fluctuations, but relatively short into the pipeline the entrainment, deposition, and gas-liquid heat transfer made the gas and liquid obtain quite similar temperatures. These results shows that the model works and runs fine also when the more advanced models like gas in liquid and energy equation are enabled.

Appendix B.5 Periodic boundaries with moving borders

This simulation simply illustrates that the model works as it should when there are moving borders and periodic boundaries (closed loop). The pipeline is 20 m long, with 0.05 m inner diameter everywhere. The initial conditions is gas everywhere, except for a slug or pig from 1 m to 2 m with initial velocity 1 m/s. Friction is turned off, and the left and right end of the slug and pig are forced to move with $C_l = 1$ and $U_0 = 0$ which ensures no mass-flux across the slug border (otherwise the slug would shed liquid and die quite fast). The simulation is run using both compressible and incompressible slugs, and a pig. In the case with a pig the density of the pig was set to the same as the slugs (1000 kg/m3). Figure 72 shows the holdup in the first section and total momentum in the system as plotted against time. It can be seen that the momentum is conserved within a small numerical error, and that the three different simulations with compressible and incompressible slugs and pigs give the same result.

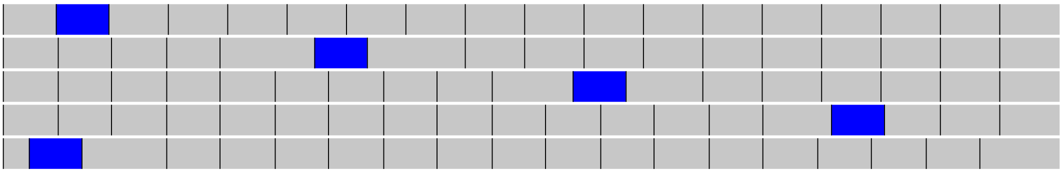


Figure 71: Periodic system with a single slug. The images shows the simulation after 0, 5, 10, 15 and 20 seconds respectively from top to bottom.

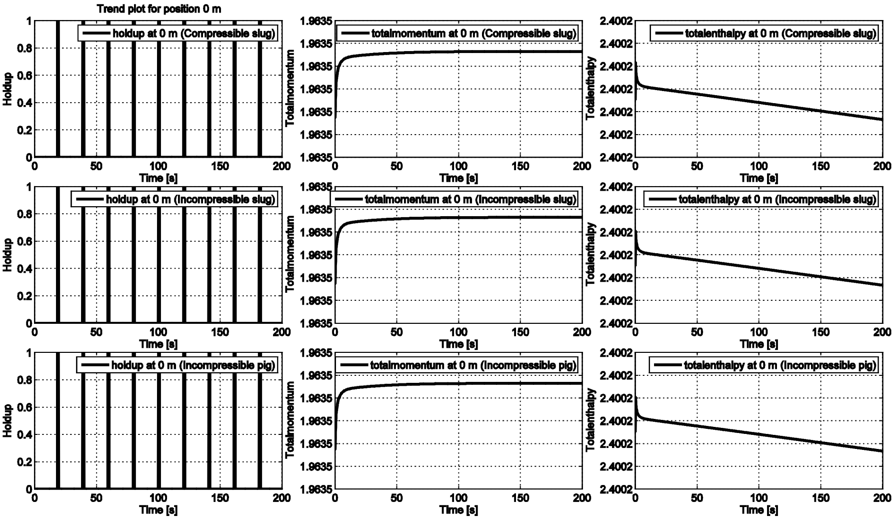


Figure 72: Holdup and total momentum as function to time at position 0 m in a periodic system. The three rows correspond to the case with a compressible slug, an incompressible slug, and pig respectively.

Appendix B.6 Symmetry test number 2

This section presents two simulations of terrain slugging in a s-shaped riser. The first simulation is run using a mass flow boundary at the inlet and a pressure boundary at the outlet, while the second case is run with a mass flow boundary at the outlet and a pressure boundary at the inlet, with the geometry reversed. The pressure boundary was set to atmospheric pressure, while mass sources with mass rates of 0.295 kg/s and 0.0021 kg/s were used for the oil and gas respectively. The two simulations are illustrated in Figure 73 and Figure 74, and a comparison of the pressure in the buffer tank between the two simulations is presented in Figure 75. It can be seen that the pressure signals are quite similar, and follow the same trend with approximately the same slugging period. The small difference most likely comes from the grid management. The grid management should be symmetric, but since the sections and checked from left to right in both cases slightly different management operations might happen (like a section being merged to the left instead of the right). It was also tested increasing the gas mass rate to 0.021 kg/s, which resulted in only stratified or annular flow. The results then became identical as no grid management operations happened.

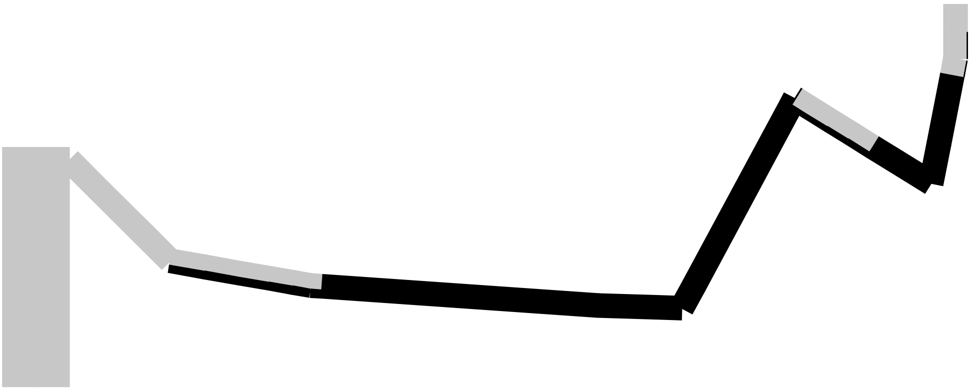


Figure 73: Terrain slugging in a s-shaped riser (time = 100 seconds).

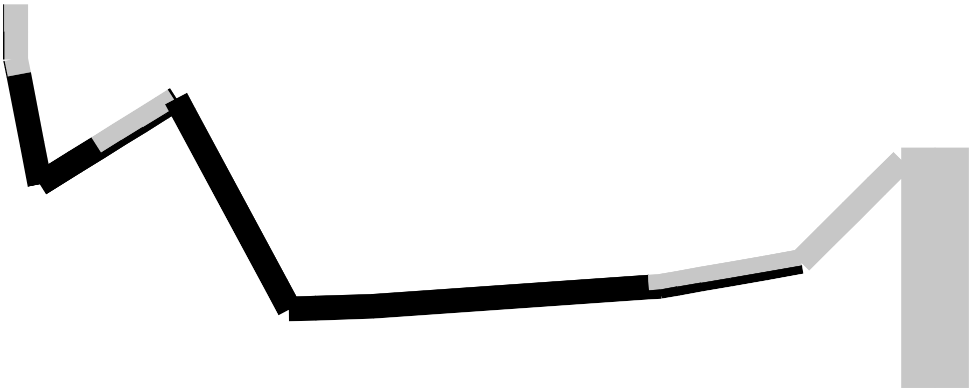


Figure 74: Terrain slugging in a s-shaped riser, reversed direction (time = 100 seconds).

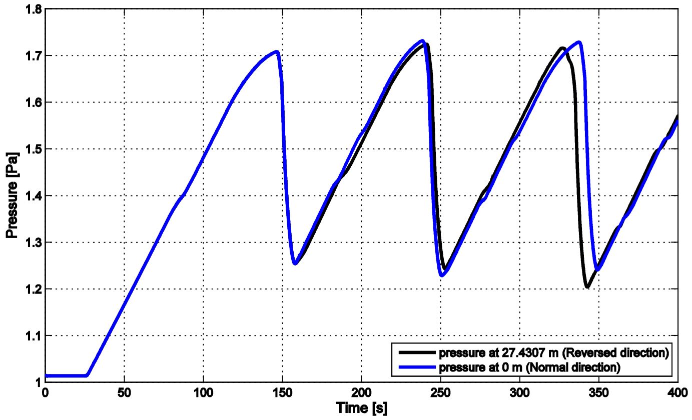


Figure 75: Comparison of the pressure in the buffer tank between the simulations of the s-shaped riser in the normal and reversed direction.

Appendix C Additional forces

Appendix C.1 Perturbations

It is possible to add perturbation forces at multiple locations in a simulation. This can be useful for instance if using a fine grid to try to capture the growth of waves, by initiating some disturbances that will either die, or grow if the conditions for growth are correct. In real-life system there will typically always be some disturbance, amongst others from turbulence, so adding such perturbation forces should be physically ok to do as long as the force is not too large. Note that the perturbation is only applied to the liquid phase.

The level gradient term is used to give the order of magnitude of the perturbation force. By using the mass and average liquid height in a momentum control volume, the perturbation force is calculated by:

$$F_{pert} = M_k g \frac{\overline{h}_l}{\Delta x} \Pi_{rand} \mathrm{X} \tag{226}$$

Here the number Π_{rand} is a random number between -1 and 1, X is a fraction between 0 and 1, $\overline{h}_l$ is the average liquid height and Δx is the length of the control volume.

A custom random number generator (a simple linear congruential generator) has also been created and is used to generate the random numbers for the perturbations. This is done to be able to save and restore the state of the generator, for instance when re-evaluating the time step or if stopping and restarting a simulation.

Appendix C.2 Valves

Valves can be added to the simulation. Currently the valve works by specifying a constant K-factor, which determines the pressure loss as shown in equation (228).

$$\Delta p_{valve} = \frac{1}{2}\rho_k K u_k^2 \tag{227}$$

The valve force is included in the momentum equation related implicitly to the new velocity as shown in equation

$$F_{valve} = V_k^n \frac{\Delta p_{valve}}{\Delta x} = \frac{1}{2\Delta x} M_k^n K \left|u_k^n\right| u_k^{n+1} \tag{228}$$

Appendix C.3 Expansion and contraction losses

When the flow enters either a contraction or an expansion in the pipe, additional losses can be added to the momentum equation. This is done with the use of *K*-values, similar to what is done for valves. Equation (230) and (231) show the *K*-values for expansion and contraction losses, respectively [86, 87].

$$K_{\text{expansion}} = \left(1 - \left(\frac{D_1}{D_2}\right)^2\right)^2 \tag{229}$$

$$K_{\text{contraction}} = 0.5\left(1 - \left(\frac{D_2}{D_1}\right)^2\right) \tag{230}$$

In equation (230) $D_1 > D_2$, while in equation (231) $D_1 < D_2$.

Note that a perfect sudden expansion or contraction is assumed here (instant transition from D_1 to D_2).

The loss from either equation can be calculated similar to the valve losses, from:

$$\begin{aligned}\Delta p_{\text{minorlosses}} &= \frac{1}{2}K\left|u_{k,D,\min}^{n}\right|u_{k,D,\min}^{n+1} \\ F_{\text{minorlosses}} &= \frac{1}{2\Delta x}M_{k}^{n}K\left|u_{k,D,\min}^{n}\right|u_{k,D,\min}^{n+1}\end{aligned} \tag{231}$$

Note that the loss is defined using the velocity in the part with the smallest diameter. Relating the velocity in the small diameter part to the average velocity in the momentum control volume by $\overline{A}u_k = u_{k,D,\min}A_{D,\min}$, we get:

$$F_{\text{minorlosses}} = \frac{1}{2\Delta x}M_{k}^{n}K\left|u_{k}^{n}\frac{\overline{A}}{A_{D,\min}}\right|u_{k}^{n+1}\frac{\overline{A}}{A_{D,\min}} \tag{232}$$

Appendix C.4 User defined forces

It is also possible to add user defined forces that will be added to the momentum equation. This can for instance be necessary when performing a simulation with periodic boundary conditions, as one will then need a driving force to move the fluids. If a force $\left(\frac{\partial \mathrm{P}_{userforce}}{\partial x}\right)$ is specified, it will be applied to both the mixture field as follows:

$$F_{userforce} = V_k \frac{\partial \mathrm{P}_{userforce}}{\partial x} \tag{233}$$

4 Papers

Paper 1

The effect of boundary conditions and droplet entrainment on severe slugging using a Lagrangian slug tracking model

Ivar Eskerud Smith, Ole Jørgen Nydal

International Journal of Multiphase Flow
Volume 85, October 2016, Pages 245–257

International Journal of Multiphase Flow 85 (2016) 245–257

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmultiphaseflow

The effect of boundary conditions and droplet entrainment on severe slugging using a Lagrangian slug tracking model

Ivar Eskerud Smith*, Ole Jørgen Nydal

Department of Energy and Process Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway

ARTICLE INFO

Article history:
Received 1 March 2016
Accepted 17 June 2016
Available online 23 June 2016

Keywords:
Gas–liquid pipe flow
Severe slugging
Droplet entrainment
Closed loop
Slug tracking

ABSTRACT

This study presents a Lagrangian slug tracking model for prediction of slug flow in multiphase flow pipelines. Simulations are compared with severe slugging experiments from the SINTEF Large Scale Multiphase Laboratory with 0.189 m inner diameter and a length of about 1000 m. The fluids used in the experiments are nitrogen and naphtha, and the system pressure is 25 bar. The model is shown to give fast and accurate results within the experimental accuracy without the need for a sub-grid model, as long as the grid size is not too large around the riser bottom where the liquid blockage occurs. It is also shown that simulations using a constant pressure boundary condition at the outlet are not always adequate, as the pressure fluctuations in the receiving facilities can significantly affect the flow. Furthermore, we show that the inclusion of a droplet field has a significant impact on the predicted slugging periods.

1. Introduction

Multiphase flow is the flow of two or more phases in the same channel/pipe. The behavior of the flow can take many forms (flow patterns), depending on several parameters like fluid velocities, pipe diameter, pipe inclination, and the fluid properties. The fluid properties are again dependent on the pressure and temperature changes in the system, especially the gas density and the fluid viscosity. The simplest of the flow patterns is *stratified smooth flow*, where the gas and liquid are separated by a clear and flat interface. Increasing the fluid velocities will increase the interfacial friction between the phases and result in waves forming on the interface. This flow pattern is referred to as *stratified wavy flow*. Depending on the conditions, these waves might become unstable, and grow large until they bridge the entire pipe cross-section and create a liquid plug. The gas pressure can then build up behind the liquid plug and accelerate it to high velocities through the pipe. This flow pattern is called *hydrodynamic slug flow* (sometimes just referred to as *slug flow*), and consists of alternating gas regions (*Taylor bubbles*) and liquid plugs (*slugs*). When a liquid blockage of the pipe cross-section occurs because of the pipe geometry, the flow is referred to as *terrain-slugging*. Liquid then typically accumulates in low-points due to the pipe elevation. When liquid slugs accumulate at the bottom of a riser, the slug might grow depending on the gas velocity and upstream gas compressibility. When the pressure overcomes the hydrostatic pressure of the liquid in the riser, the gas will start to push the slug through the riser, and when the Taylor bubble enters the riser the slug is violently blown out. This type of slugging is called *severe slugging*, or *riser-induced slugging*. It is important to be able to predict the frequency and liquid volume of such severe slugs, especially for the design of slug catchers in receiving facilities so that they are not overfilled.

For a slug being pushed from the left to the right by the preceding Taylor bubble, the velocity of the left end of the slug (*bubble nose*) will be that of the nose of the Taylor bubble. The right end of the slug typically absorbs liquid (*slug front*) from the liquid below the upstream Taylor bubble, but this depends on several parameters like the difference in liquid velocity between the slug and the film below the Taylor bubble. It might happen that both ends of a slug are fronts absorbing liquid, or both ends might be Taylor-bubble noses. Empirical correlations to predict the velocity of a Taylor bubble nose has been extensively investigated; see for instance (Bendiksen, 1984; Benjamin, 1968; Dumitrescu, 1943; Jeyachandra et al., 2012; Joseph, 2003). In the two-fluid model the slug velocity is a result of a combined effect of among others friction models and the numerical scheme. Previous work (Kjølaas et al., 2013) has shown how the interfacial friction models in the commercial simulator LedaFlow (Kongsberg Oil and Gas Technologies, 2015; SINTEF, 2015) had to be tuned at high holdup values to obtain slug velocities that corresponds with well-established slug velocity correlations (Bendiksen, 1984).

This study presents the performance of a Lagrangian slug tracking model, where the slug control volume velocity is set according to well established bubble nose velocity correlations. This

* Corresponding author.
E-mail addresses: ivar.eskerud@gmail.com (I.E. Smith), ole.j.nydal@ntnu.no (O.J. Nydal).

http://dx.doi.org/10.1016/j.ijmultiphaseflow.2016.06.014
0301-9322/© 2016 Elsevier Ltd. All rights reserved.

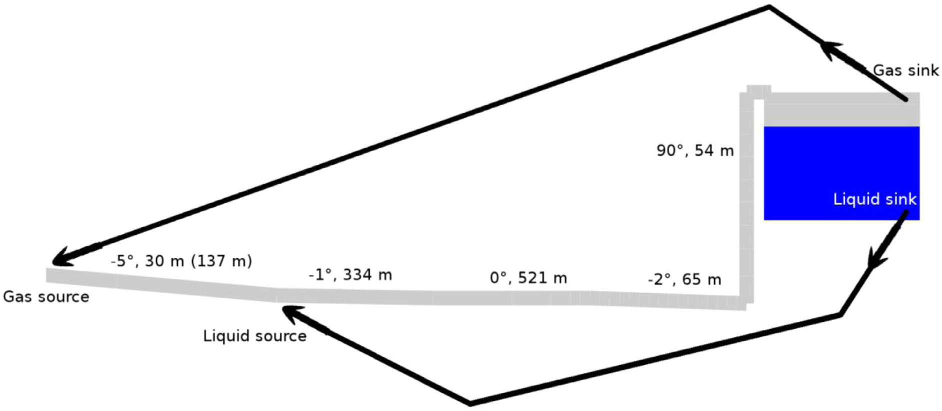


Fig. 1. Illustration of the geometry used in the simulations (not correct scale).

Table 1
Pipeline geometry for the simulations. The inner diameter used in the simulations is 0.189 m for all pipes, and the hydraulic roughness is 29 µm. The exception is the separator which was modeled with an inner diameter of 2 m.

Length [m]	30*	334	521	65	54	32.85
Cumulative length (m)	30	364	885	950	1004	1036.85
Inclination (°)	−5	−1	0	−2	90	−90

* About 107 m of pipe is added to the first 30 m long section, to include the volume present in the gas feed line.

automatically gives the desired velocities for the slugs, and enables sharp gas-liquid fronts at the slugs with reduced numerical diffusion. This allows for the use of a coarse grid, giving fast simulations without the need for a sub-grid model like the unit-cell model concept (Dukler and Hubbard, 1975) typically used by commercial simulators. The model presented in this paper is an extension of the model framework described in Kjeldby (2013).

2. Experiments

2.1. Experimental facility

Experiments were carried out in the SINTEF Large Scale Multiphase Laboratory in 1988, using nitrogen and naphtha as fluids at a nominal pressure of about 25 bar. The data are presented with permission from SINTEF. The flow loop consists of pipes with both 0.189 m and 0.194 m inner diameter, but for simplicity 0.189 m has been used for all pipes in the simulations. The pipeline consists among others of a 522 m long horizontal section, followed by a 65 m long -2° inclined part, and a 54 m tall riser. A smaller separator ($\sim$7 m^3) is located at the top of the riser, before the fluids are returned to the main separator through the 0.6 m inner diameter drop-leg. The total length of the flow loop is about 1000 m. In the simulations the volumes of the smaller separator and the drop-leg has been combined with the large separator. The pipeline geometry used in the simulations is listed in Table 1, and illustrated in Fig. 1. The total gas volumes upstream and downstream the riser are shown in Table 2.

2.2. Fluid properties

The fluid properties for the naphtha and nitrogen used in the experiments are shown in Table 3.

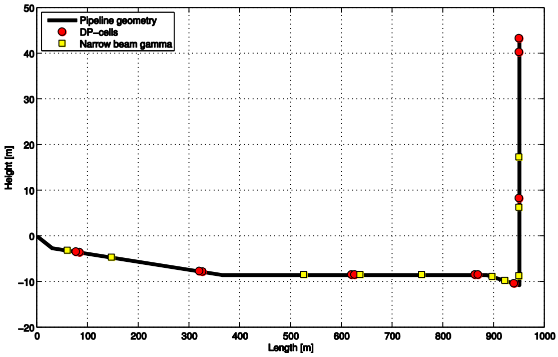


Fig. 2. Positions of gamma densitometers and dp-cells in the pipeline. The gas feed line is not included, so that the mixing point of the gas and liquid is located at the coordinates (0,0).

Table 2
Relevant gas volumes in the experimental facility.

Section	Volume (m^3)
Upstream mixing point	3
Flowline upstream riser bottom	26.6
Downstream riser	51.6

Table 3
Fluid properties for nitrogen saturated naphtha and nitrogen at 20 bar and 30 °C.

	Density (kg/m^3)	Viscosity (cP)	Gas–liquid surface tension (mN/m)
Liquid (naphtha)	680	0.28	15.8
Gas (nitrogen)	23.6	1.7e-2	

The nitrogen density is modeled after a linear correlation given in the experimental report from SINTEF, and is shown in Eq. (1):

$$\rho_g = 0.49 + 1.1532p \tag{1}$$

Here p is the pressure in bar, and the resulting gas density is in kg/m^3. The other fluid properties were modeled as constant values (thus incompressible liquid). The temperature was kept at approx-

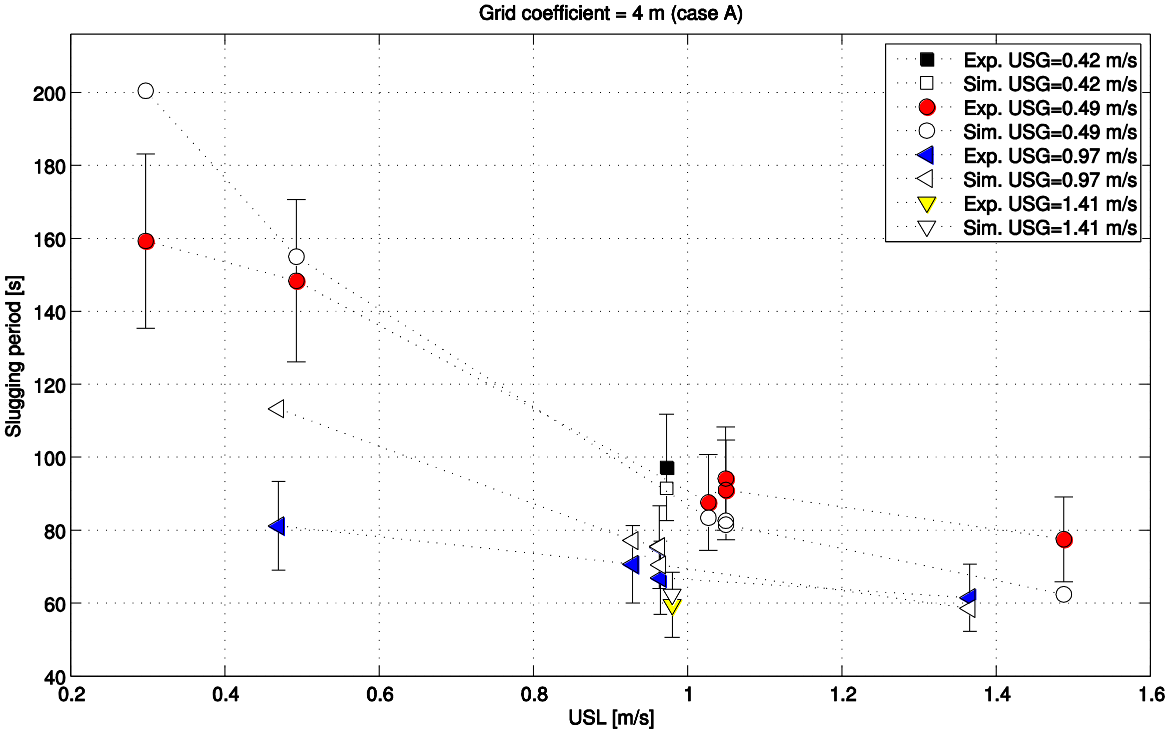


Fig. 3. Comparison of the average slugging period between the experiments and the simulations with 4 m grid size around the riser (case A), plotted against USL for different values of USG. The simulations are performed with a constant pressure boundary. The error bars represent 15% deviation.

imately 30 °C in all experiments, and varied typically with at most ±1.5 °C within an experiment.

2.3. Instrumentation

The gas and liquid flow rates were measured with vortex meters and turbine meters respectively. Multiple vortex meters and multiple turbine meters were used, coupled in parallel in the feed lines. The gas and liquid flow meters were calibrated to give superficial gas and liquid velocities (USG and USL) directly as output instead of volumetric flow rates. The gas and liquid mass rates to be used in the simulations were calculated from the measured superficial velocities by using the fluid densities calculated from pressure and temperature measurements at the flow meters and the cross sectional pipe area. Each experiment was run with a constant gas and liquid mass flow rate. Automatic control valves were installed in the feed lines to try to keep the flow rate constant during the subsequent pressure buildup and blowout in the severe slugging cycles. Unfortunately these valves did not work as well as expected, which resulted in an increase in the flow rates during blowout. The maximum value of both the gas and liquid flow rates typically varied with about 5–15% from the average flow rate. The flow rates used in the simulations were chosen as the average value one would get if removing these peaks. This has to be regarded as an uncertainty in the experiments, but how large uncertainty this translates to for the slugging period is not known.

Both holdup and pressure were measured at multiple locations along the pipeline. The holdup was measured using both fast volume weight meters and narrow beam gamma densitometers, giving in total 13 holdup measurements distributed along the pipeline. The pressure was measured by differential pressure cells (dp-cells) connected to a common reference line filled with nitrogen, so that the absolute pressure at any location could be obtained by subtracting the differential measurement from the absolute pressure in the reference line. In total 13 dp-cells were distributed along the pipeline. The locations of the gamma densitometers and dp-cells are shown in Fig. 2.

The uncertainty for the narrow beam gamma densitometers is not reported, but from other experimental campaigns the repeatability of the narrow beam gammas has been reported to be around 0.02–0.03 in absolute value. The actual uncertainty might be even higher than this for intermediate holdup values, so it is estimated that the uncertainty can be as large as 0.05 in absolute value. The holdup measurements are however not of primary interest in this study, as the focus is on the prediction of the severe slugging period. The uncertainty of the dp-cells was 0.1% of the range, approximately 0.003 bar (Table 4).

Table 4
Measurement uncertainties.

Instrument	Uncertainty
USG (vortex meters)	0.65% of measured value
USL (turbine meters)	0.6% of measured value
Narrow beam gamma densitometer	0.05 (absolute value)
Differential pressure cell	0.003 bar (0.1% of range)

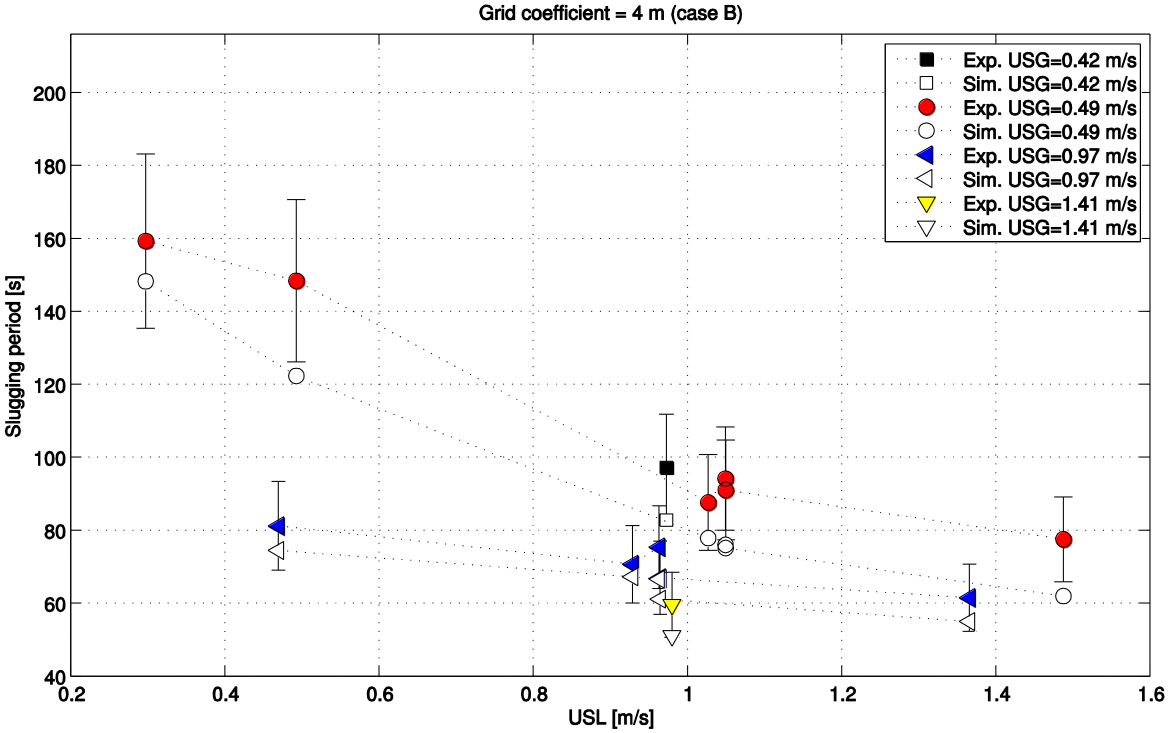


Fig. 4. Comparison of the average slugging period between the experiments and the closed loop simulations with 4 m grid size around the riser (case B), plotted against USL for different values of USG. The error bars represent 15% deviation.

3. Simulations

The numerical model is described in detail in Appendix A.

The experiments are simulated both using an open outlet with a constant pressure boundary condition, and as a closed loop system. One of the main concerns in the presented simulations is what pressure to use for the constant pressure boundary, since we know that the pressure in the separator will fluctuate with the severe slugging. This is simply a limitation of simulations using this type of boundary condition when simulating this type of experiments. As will be shown, it is not always possible to get correct results with a constant pressure boundary condition for the type of experiments that is presented in this study.

In the closed loop system the gas and liquid exit the separator by the use of mass sinks, with the same mass flow rates that are fed into the start of the pipeline as shown in Fig. 1. The total available volume in the separator and other process facilities is reported to be 51.6 m^3. However in the simulations the separator was made twice as large (103.2 m^3), and filled initially with 75% liquid. The reason for this is to avoid that the liquid mass sink in the separator is running out of liquid. The exact amount of liquid present in the system was not known, but was found by varying the initial amount of liquid in the separator until the pressure amplitudes both in the separator and before the riser corresponded with the experiments. An initial liquid holdup of 75% in the separator showed to give good results for all experiments, indicating that the total amount of liquid present in the system is constant. The amount of liquid present in the separator at steady state in each experiment will however vary, which also means that the available compressible gas volume in the separator will vary. To include the volume of the gas feed line (3 m^3) in the simulations, an extra length of about 107 m was added to the first 30 m long -5° inclined section. The total volume upstream the bottom of the riser is then approximately 29.6 m^3. An overview of the different boundary conditions used in the simulations are shown in Table 5.

Table 5
Overview of boundary conditions used in simulations.

Case type	Inlet border boundary	Outlet border boundary
Open outlet	Mass flow specified	Constant pressure in outlet section (ghost section to the right of the outlet border). The unknown velocity in the outlet section is extrapolated from the outlet border.
Closed loop	Mass flow specified	Mass flow specified

3.1. Boundary conditions

The constant pressure at the open outlet boundary was set equal to the average experimentally measured pressure in the separator, while the velocity in the outlet section (ghost cell) was calculated by extrapolating the velocity in the last cell. The liquid holdup in the outlet section was constant and set to 0 in the simulations with an open outlet. Note that the separator is not included in the simulations with the open outlet boundary. The open outlet rather represents the entrance to the separator, which has a fixed pressure. It should also be noted that when a mass flow boundary was used for the inlet border and outlet border, the mass flow

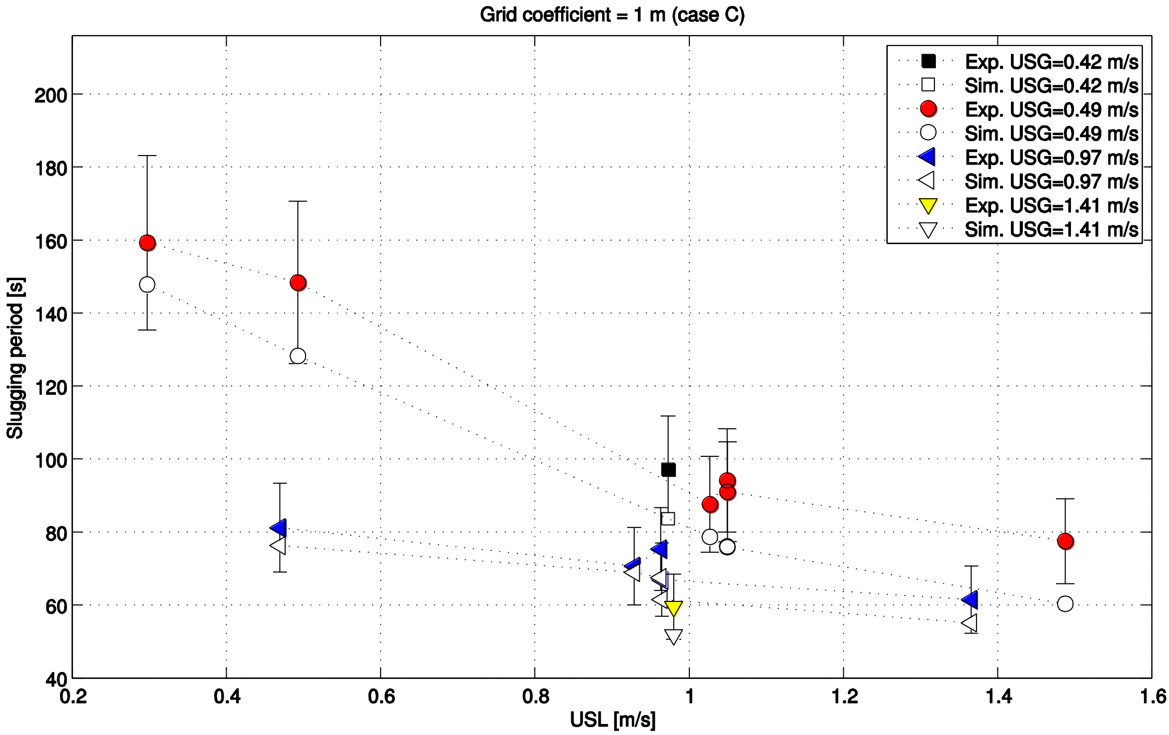


Fig. 5. Comparison of the average slugging period between the experiments and the closed loop simulations with 1 m grid size around the riser (case C), plotted against USL for different values of USG. The error bars represent 15% deviation.

were in all cases set to 0. Instead the fluids enter and leave the system through mass sources and mass sinks, at the locations marked in Fig. 1. This was done among others to make the liquid enter after the gas feed line, and to be able to remove gas from the top of the separator as the gas phase in the separator is not in contact with the outlet border (which in the simulations is at the bottom of the -90° inclined separator).

3.2. Grid size

A large grid size (grid coefficient $\Delta x_{\text{bubble}} = 100\,\text{m} = 529\,\text{D}$) was used for the first 885 m, while a finer grid was used in the -2° inclined section before the riser and in the riser, to be able to capture the liquid buildup and subsequent blockage. Simulations were run with both 4 m and 1 m grid coefficients at these locations. For an explanation of the numerical grid see Eq. (10) in Appendix A.1. All cases were also run both with and without liquid droplet entrainment enabled. It is well known that correct modeling of the droplet field can be of high importance in multiphase flow simulations, but this study specifically shows the effect the droplet field has on severe slugging. The case with an open outlet with a constant pressure boundary condition was only run with a 4 m grid size. An overview of the simulation setups are shown in Table 6.

Note that a 4 m grid coefficient means that the grid size can vary from 4–8 m, as described by Eq. (10). The corresponding range for the 1 m grid coefficient is 1–2 m, and 100–200 m for a 100 m grid coefficient.

The closed loop simulations were as mentioned started with the separator filled with 75% liquid. When starting a simulation the liquid will start to distribute throughout the pipeline, and after some time the pressure fluctuations will reach a statistically steady state condition with steady amplitude and slugging period. The experiments were reported to been run with a minimum of 45 min stabilization time. All simulations were run for 5000 s (almost 84 min), and the last 1000 s were chosen for comparison with the slugging period from the experiments measured at the inlet. Shorter stabilization times should be expected with increasing superficial mixture velocity, but for simplicity all simulations were run for the same amount of time. The simulations that were simulated with an open outlet were also simulated for the same amount of time, with the same stabilization time.

Table 6
Grid coefficients used in the different simulations.

Case name	First 885 m	65 m long -2° inclined part before riser	Riser	Closed loop
A	100 m (529 D)	4 m (21 D)	4 m (21 D)	No
B	100 m (529 D)	4 m (21 D)	4 m (21 D)	Yes
C	100 m (529 D)	1 m (5 D)	1 m (5 D)	Yes

4. Results without droplet entrainment

The slugging period from the simulations with an open outlet (case A) are compared to the experimental values in Fig. 3. It can be seen that the slugging period is significantly overestimated for the experiments with lowest USL. For higher values of USL the slugging period is predicted within 15% deviation for most experiments, though somewhat underpredicted for the largest values of USL.

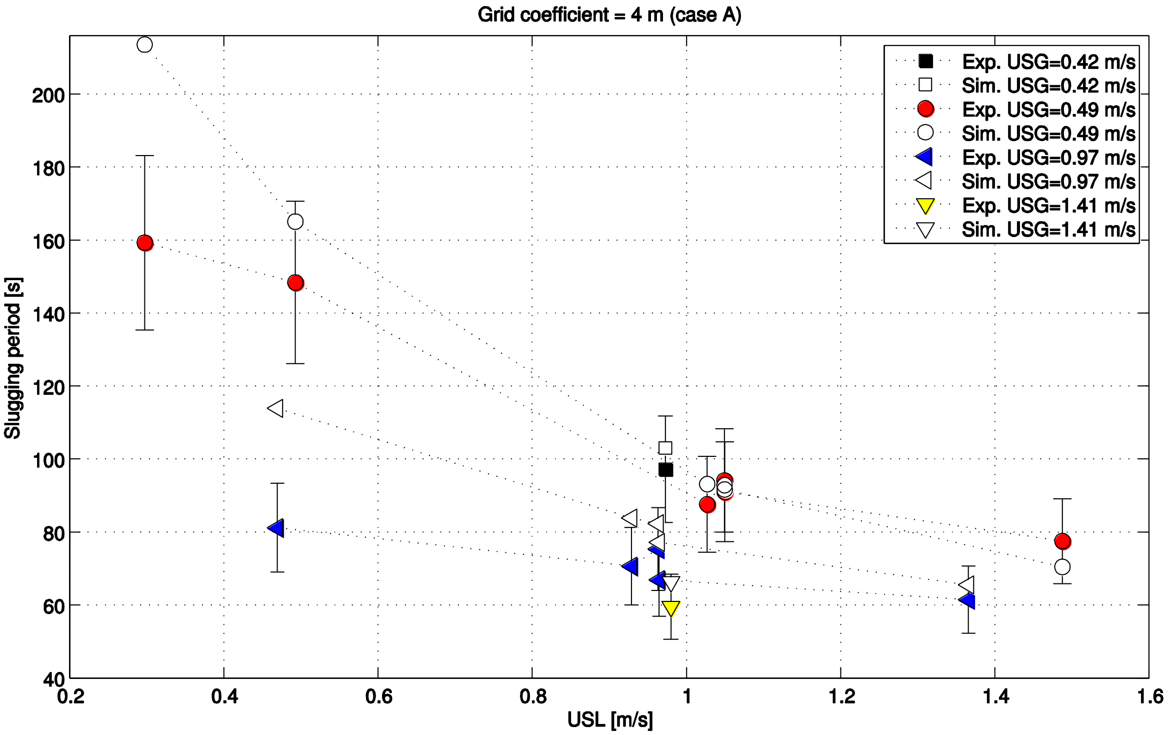


Fig. 6. Comparison of the average slugging period between the experiments and the simulations with 4 m grid size around the riser (case A), plotted against USL for different values of USG. The simulations are performed with a constant pressure boundary, and with liquid droplet entrainment enabled. The error bars represent 15% deviation.

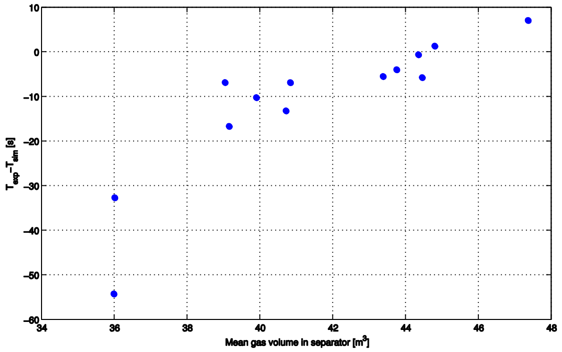


Fig. 7. Difference in slugging period (T) between experiments and the case with a constant pressure boundary and entrainment enabled, plotted against the mean available gas volume in the separator. The gas volume is obtained from the closed loop simulations (case C).

To improve the results, the experiments were simulated as a closed loop system as previously described. Fig. 4 shows the comparison of the average slugging period from the experiments and simulations, with a 4 m grid coefficient around the riser (case B). These results show that though some of the experiments are predicted within the estimated uncertainty, the slugging period is systematically under-estimated for most experiments. The slugging periods for some of the lowest USL values are however significantly improved in comparison with the simulations with an open outlet. In general, only a few of the experiments can be said to be well predicted.

To check if the results were improved with a finer grid, the simulations were also run with a 1 m grid coefficient around the riser area (case C). These results are shown below in Fig. 5.

The results show a similar trend as for the 4 m grid case: The slugging period is systematically underpredicted, especially for low values of USG. In general, the closed loop simulations without entrainment show a too low slugging period. A too low slugging period means that the liquid blockage at the bottom of the riser occur too early, so that the pressure buildup starts too early. It was hypothesized that the liquid transport out of the riser was too small, leading to too much liquid falling back down the riser and creating a liquid blockage too early. This will be investigated in the following section.

4.1. Results with droplet entrainment

Liquid droplet entrainment was included in the model, to test if this would remedy the problem with a too early liquid blockage. The correlations for the entrainment and deposition rates are presented in Appendix A.1.2.

The results from the simulations with entrainment with a constant pressure boundary condition are shown in Fig. 6.

For the simulations with medium and high USL the slugging periods are predicted quite well, but the slugging periods at low USL are severely overpredicted and the results compare even less favorable than the results without entrainment. Under certain conditions, it is clear that a constant pressure boundary is not adequate. In Fig. 7 the difference in slugging period between the experiments and the simulations with entrainment and a

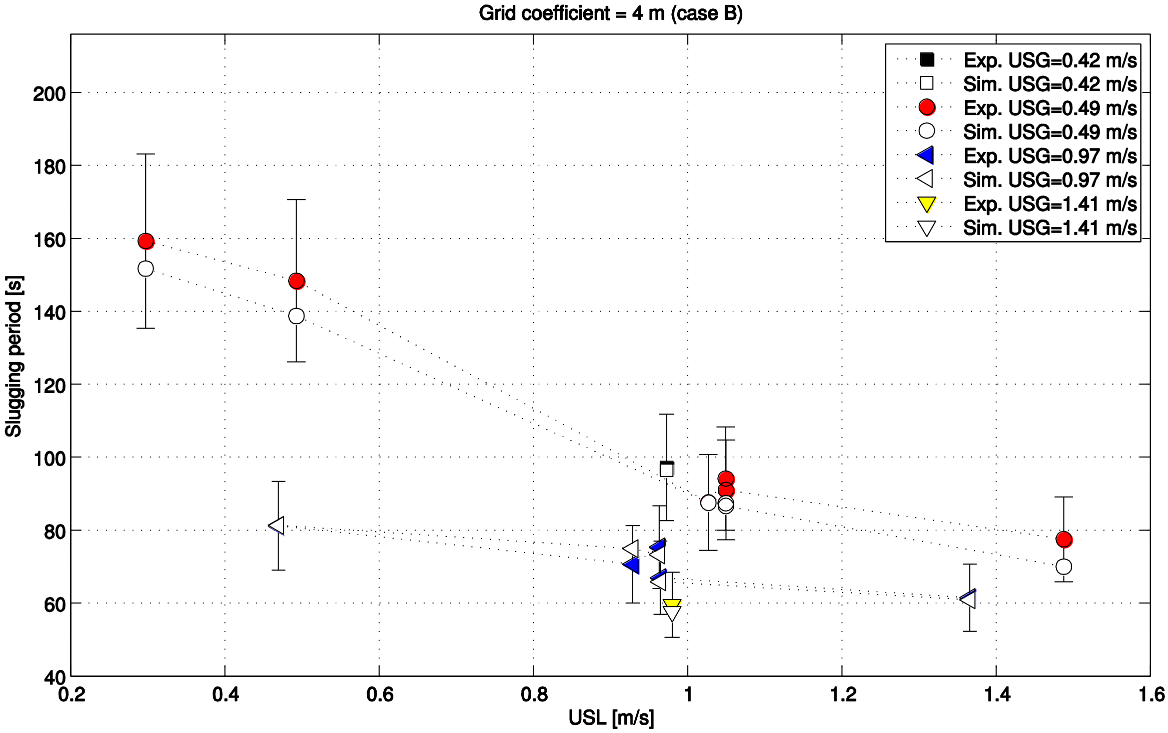


Fig. 8. Comparison of the average slugging period between the experiments and the closed loop simulations with 4 m grid size around the riser (case B), plotted against USL for different values of USG. Liquid droplet entrainment is enabled. The error bars represent 15% deviation.

constant pressure boundary are plotted against the mean available gas volume in the separator (obtained from the corresponding closed loop simulations with entrainment). One can clearly see that the deviation in slugging period between the simulations and experiments increases as the available compressible volume in the separator is decreasing. The reason for this is most likely that the pressure in the separator will increase faster during blowout when the available compressible gas volume is smaller. The increasing separator pressure acts as a decelerating force on the slug being blown out, and the blowout process stops earlier. In addition, after the riser bend has been blocked, gas is drained from the separator while no or little gas is allowed to pass the liquid blockage. This will result in a decreasing pressure in the separator during the buildup period, which acts as a suction force on the slug in the riser. These two effects are not captured when using a constant pressure boundary, which results in a longer blowout time and too large pressure amplitudes and slugging periods.

The results for the closed loop simulations with entrainment are shown below in Figs. 8 and 9 respectively.

The results shown in Figs. 8 and 9 indicate that including liquid droplet entrainment improves the predicted slugging periods for the closed loop simulations, most likely due to more liquid being transported as droplets in the gas phase during blowout. The results are more or less independent of the grid size used in the simulations. Though the improvement is not as large as the improvement found from using the correct boundary conditions, a systematic improvement is seen for all experiments (comparing Figs. 4 and 5 with Figs. 8 and 9). The average deviation from the experimental slugging period for simulation cases B and C decreases from about 12% without entrainment, to about 4% with entrainment.

Comparing the results with a closed loop system with the results with a constant pressure boundary there is a significant improvement of the predicted slugging periods, especially for the experiments with the lowest values of USL. These are the experiments with the smallest amount of gas in the separator shown in Fig. 7. The predicted and experimental pressure fluctuations at the inlet for the experiment with USL = 0.47 m/s and USG = 1 m/s is shown in Fig. 10, and clearly shows the difference in results between a constant pressure boundary and using a closed loop system.

Including entrainment also acts as increased interfacial friction due to momentum exchange between the gas and liquid, accelerating the liquid. It is therefore possible that a modified interfacial friction correlation could have given a similar improvement in the results, in combination with including droplet entrainment. It is however difficult to quantify the relative contributions of the interfacial friction and the entrainment from the available experimental data. The entrainment was not measured in the experiments, but comparing the holdup at the top of the riser between the simulations and experiments indicates that the hypothesis that too much liquid falls back down the riser due to the lack of a droplet field is correct. Fig. 11 shows the holdup at the top of the riser for a selected experiment, showing the simulation results both with and without entrainment. It is clear that both the predicted holdup and slugging period are significantly improved by adding a droplet field. With a droplet field the holdup becomes lower during each blowout, meaning that more liquid is transported out of the riser as hypothesized.

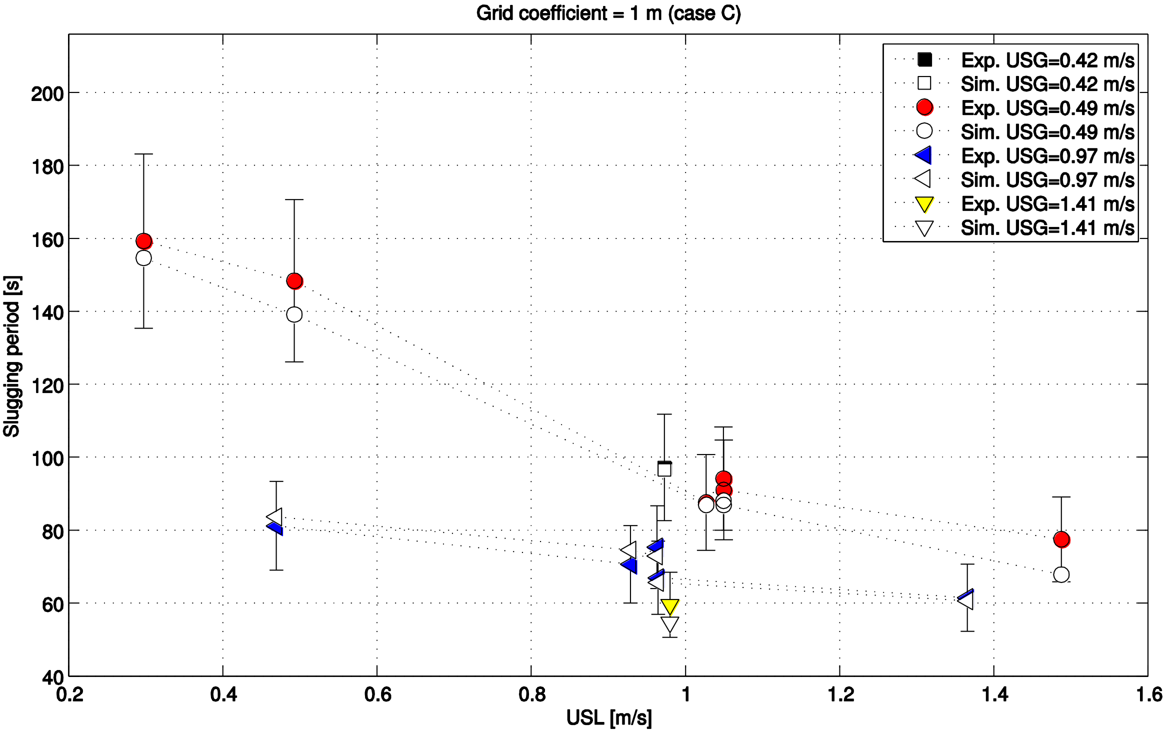


Fig. 9. Comparison of the average slugging period between the experiments and the closed loop simulations with 1 m grid size around the riser (case C), plotted against USL for different values of USG. Liquid droplet entrainment is enabled. The error bars represent 15% deviation.

4.2. Simulation speed

The results presented in this paper shows that the use of a Lagrangian slug tracking model can give good predictions of the severe slugging period, also when using a relatively coarse grid. The simulation speed of the current model is presented in Fig. 12. The simulations were run on a laptop computer with 16 GB RAM and a 2.4 GHz Intel i7-4900MQ CPU. The CPU time (time to run a simulation) is dependent on many other factors than just the specifications of the RAM and CPU, but at least give an illustration of the order of magnitude of the CPU time compared to real time. The volume fraction error tolerance for solving the mass and momentum equations is 0.001 in all simulations, the same as used in all simulations in this study.

For the simulations with the lowest superficial mixture velocities the CPU time is about 900 times faster than real time, meaning that simulating 5000 s of real time would take approximately 5.5 s of computational time. The CPU time increases with increasing mixture velocity, as the gas and liquid velocities in the pipeline becomes higher resulting in a lower time step from the CFL criterion. As the time until stable pressure oscillations are reached decreases and the slugging frequency increases with increasing superficial mixture velocity the CPU time could have been decreased, but for simplicity all experiments were run for the same amount of time.

5. Conclusions

Simulations of a pipeline riser system using a Lagrangian slug tracking model have been presented. With the correct boundary conditions and inclusion of liquid droplet entrainment, the presented model is shown to give fast and accurate predictions of the severe slugging period. The results show that modeling the outlet as an open boundary with a constant pressure can lead to the slugging period being severely overpredicted, especially when the available compressible gas volume in the receiving facilities becomes small. The decelerating pressure force from the gas in the separator on the slugs being blown out is then not captured, and the blowout period will last too long. The importance of including liquid droplet entrainment is illustrated, which can significantly affect the amount of liquid being transported out of the riser during blowout. Liquid droplet entrainment also acts as increased interfacial friction, which also plays an important role in accelerating the liquid in the riser during blowout. Together the combined effect or entrainment and increased interfacial friction in the riser increases the amount of liquid being transported out during blowout. This decreases the amount of liquid falling back down the riser, and leads to a longer time before liquid blockage in the riser bottom occur and a new pressure buildup period starts. Without a droplet field the liquid blockage occur too early, and the slugging period is predicted too low.

Acknowledgments

The authors acknowledge Shell Technology Norway grant no. SC12009 for their financial support of this work, and the SINTEF Multiphase Flow Laboratory for allowing us to use and publish the experimental data.

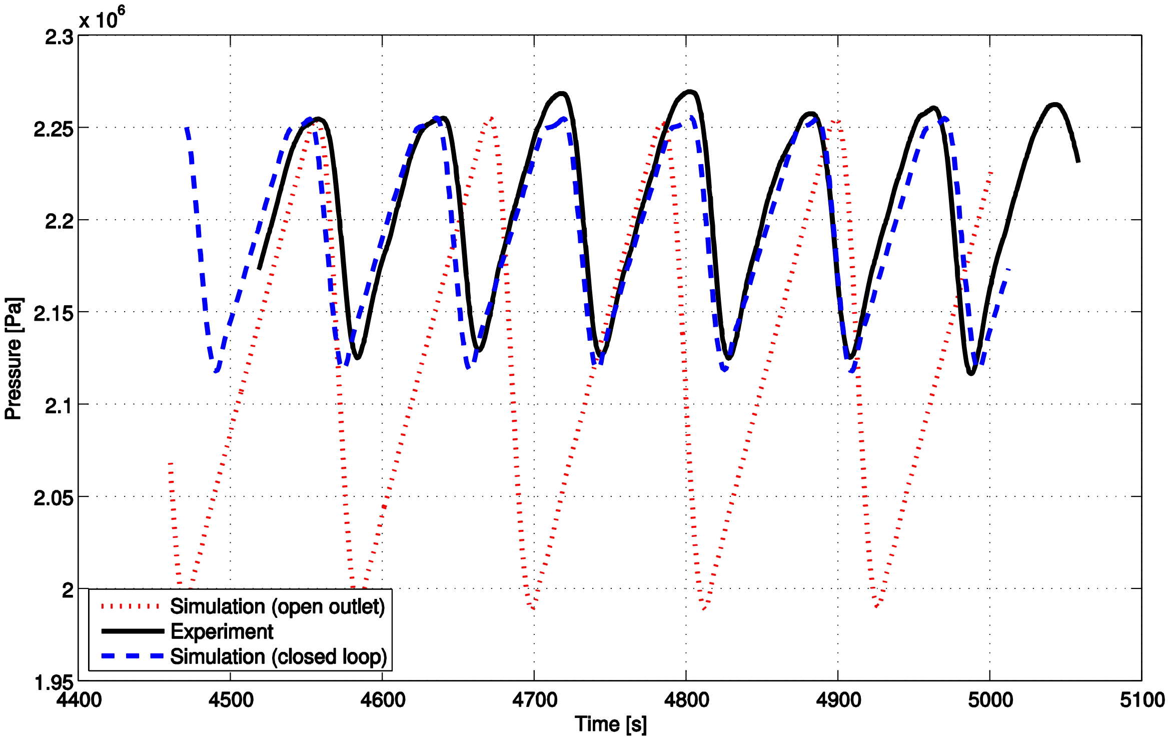


Fig. 10. Comparison of experimental and predicted slugging period for the experiment with USL = 0.47 m/s and USG = 0.97 m/s. The simulations were both run with entrainment, and with a grid size of 4 m. The pressure from the closed loop simulation and the experimental pressure have been shifted to match the first peak of the pressure signal of the open outlet simulation for better comparison.

Appendix A. Model description

The model presented in this study is a finite volume Lagrangian slug tracking model based on a previously published model concept (Nydal and Banerjee, 1996). The model is written in C++, using object oriented programming. Control volumes are represented by objects, like a bubble-section object and slug-section object. Similarly, the borders (cell-faces) between objects are represented by different types of border objects. When the liquid holdup in a bubble section approaches a user defined limit (set to 0.98 in the current study), the bubble section is converted to a slug section object.

A.1. Mass, momentum and pressure equation

The mass, pressure and momentum equations are all integrated over the gas and liquid control volumes, and are shown in Eqs. (4)–(11). A first order upwind convection scheme (Courant et al., 1952) is used where new values are needed at locations where they are not defined. These upwind values are marked by a “hat” (^) above the variable. Superscript n and $n+1$ represent current and next time index, respectively. Subscripts j denotes values defined at the control volume center, while subscripts J - 1 and J denote values at the left and right face of the control volume. Subscripts j - 1 and $j+1$ represents the values at the previous and next control volumes. The indexing convention for the momentum equation for the bubble-bubble borders and for the slugs are shown in Figs. 13 and 14 respectively. The mass equation is solved for all of the section objects regardless of type, while the momentum equation is solved for the border objects (staggered grid). For slug sections however, the momentum equation is solved over the entire slug body, and not at the slug borders. Consequently the control volumes of the neighboring bubble-bubble borders are extended all the way to the slug to make the control volumes start exactly at the end of the previous control volume.

Momentum is solved for the mixture liquid and mixture gas velocity, while the mass equation is solved for all sub-fields. By subfield is meant the individual fields that together make a mixture field, for instance oil droplets and pure gas in mixture gas field. The sub-fields are denoted by f, while the mixture fields are denoted by k. The time step is controlled by the Courant–Friedrichs–Lewy (CFL) criterion (Courant et al., 1928), modified to account for the movement of the control volume borders. This yields the following two criteria to be evaluated for all sub-field velocities at each grid cell (using the indexing convention for the borders):

$$\Delta t \leq C_{CFL} \cdot \frac{\Delta x_J}{(u_{b,j-1} - u_{f,j})}, \quad \Delta t \leq C_{CFL} \cdot \frac{\Delta x_J}{(u_{f,j} - u_{b,j+1})} \tag{2}$$

Here C_{CFL} is the user specified CFL-number for the simulation (0.8 in the current study), while u_b is the border velocity. These two criteria describe that a fluid particle located near a border should not move past the new positions of the neighboring borders within one time step. Note that these criteria should only be included if the result is positive. A negative result simply means that the border is moving away faster than the fluid, so that a fluid particle will never move past the neighboring border regardless of the time step (a negative time step would be required, which is un-

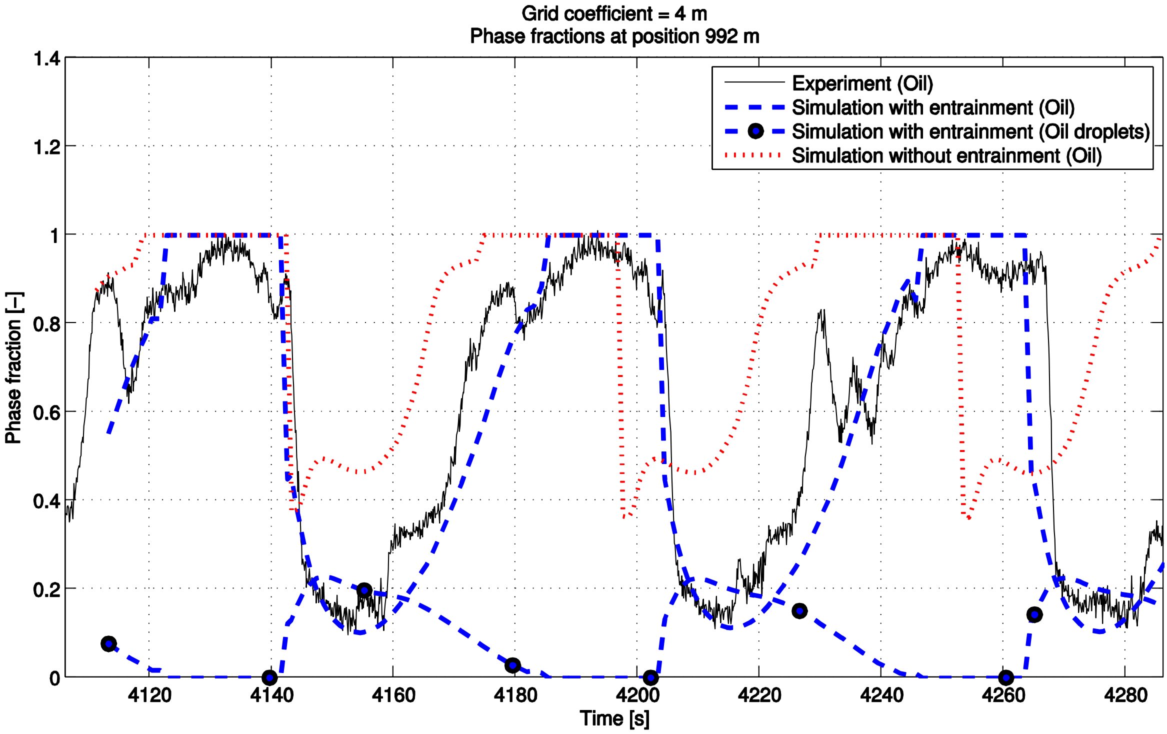


Fig. 11. Comparison of the predicted and experimentally measured holdup in the riser for a selected experiment. USG = 0.97 m/s, USL = 1.37 m/s. The simulations are run with as a closed loop with a 4 m grid coefficient (case B).

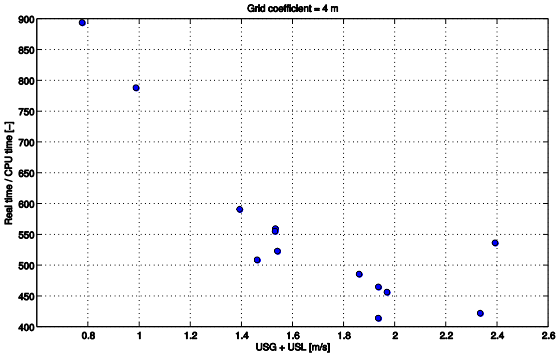


Fig. 12. Real time divided by CPU time as function of superficial mixture velocity. The simulations are all run as a closed loop, using a 4 m grid size around the riser area (case B). All simulations were run for 5000 s.

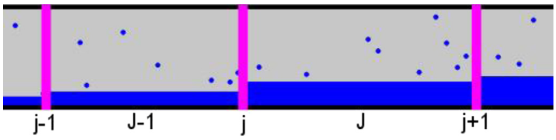


Fig. 13. Indexing convention for the momentum equation for bubble–bubble borders.

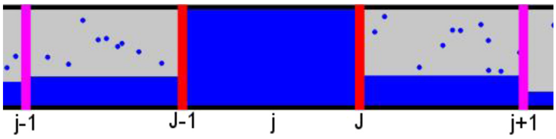


Fig. 14. Indexing convention for the momentum equation for slugs.

physical). One can also see from Eq. (2) that when the fluid velocity and border velocity approaches the same value, the time step can be arbitrary large.

The continuity/mass equation for a sub-field f is obtained from the Reynolds transport theorem (White, 2011):

$$\frac{\partial M_f}{\partial t}\oint_{A_{bf}} \rho_f(\boldsymbol{u}_f-\boldsymbol{u}_b)\cdot\boldsymbol{n}_f dA_{bf} = \dot{M}^{src}_f \qquad (3)$$

Here $\dot{M}^{src}_{f,j}$ represent the contributions from mass sources, sinks, entrainment and deposition, while A_{bf} is the boundary area and u_b the velocity of the boundary.

The discretized mass equation for sub-field f reads:

$$\frac{M^{n+1}_{f,j}-M^{n}_{f,j}}{\Delta t} = \hat{m}^{n+1}_{f,J-1}A^{n}_{J-1}\left(u^{n+1}_{f,J-1}-u^{n+1}_{b,J-1}\right) - \hat{m}^{n+1}_{f,J}A^{n}_{J}\left(u^{n+1}_{f,J}-u^{n+1}_{b,J}\right)+\dot{M}^{src}_{f,j} \qquad (4)$$

Here V_f is the fluid volume, V the total cell volume, M_f the fluid mass, A the pipe cross sectional area, u_f the fluid velocity, u_b the border velocity, while m is the specific mass defined as:

$$m_f = \rho_f\alpha_f = \frac{\rho_f V_f}{V} = \frac{M_f}{V} \qquad (5)$$

The terms ρ_f and α_f represent the fluid density and phase fraction respectively.

The pressure equation is obtained by expanding the time derivative of mass in the Eq. (3) by the product-rule:

$$\frac{\partial M_f}{\partial t} = \frac{\partial(\rho V)_f}{\partial t} = \left[\left(\frac{\partial \rho_f}{\partial p}\right)_{T_f}\frac{\partial p_f}{\partial t} + \left(\frac{\partial \rho_f}{\partial T}\right)_{p}\frac{\partial T_f}{\partial t}\right]V_f + \rho_f\frac{\partial V_f}{\partial t}$$
$$= \dot{M}_f^{src} - \oint_{A_{bf}} \rho_f\left(\boldsymbol{u}_f - \boldsymbol{u}_b\right)\cdot \boldsymbol{n}_f dA_{bf} \tag{6}$$

Eq. (6) is then divided by the density ρ_f, and summed over all sub-fields.

Using that the sum of all sub-field volumes should sum to the total cell volume, the sum of the individual volume time derivative terms becomes:

$$\sum_f \frac{\Delta V_f}{\Delta t} = \frac{\Delta V}{\Delta t} = \left(A_J^n u_{b,J}^{n+1} - A_{J-1}^n u_{b,J-1}^{n+1}\right) \tag{7}$$

Here the time derivative of the total cell volume has been expressed by the movement of the left and right cell borders. The final discretized pressure equation is shown in Eq. (8):

$$p_j^{n+1} - p_j^n = \Delta p_j$$
$$= \Delta t\,\frac{\Psi_j^n + \sum_f \frac{1}{\rho_{f,j}^n}\left[\hat{m}_{f,J-1}^{n+1}A_{J-1}^n\left(u_{f,J-1}^{n+1} - u_{b,J-1}^{n+1}\right) - \hat{m}_{f,J}^{n+1}A_J^n\left(u_{f,J}^{n+1} - u_{b,J}^{n+1}\right) + \dot{M}_{f,j}^{src}\right] - \left(A_J^n u_{b,J}^{n+1} - A_{J-1}^n u_{b,J-1}^{n+1}\right)}{\sum_f \frac{V_{f,j}^{n+1}}{\rho_{f,j}^n}\left[\left(\frac{\partial \rho_{f,j}}{\partial p}\right)_T^n\right]} \tag{8}$$

In Eq. (8) the variation of density with temperature has been left out for simplicity, as no temperature calculations are performed in the simulations since the experiments were kept at a constant temperature; besides, all the sub-fields have a common pressure. The term Ψ is the volume error correction term which is described in Eq. (12), while the last term represents the volume time derivative. Since we solve for the momentum of the mixture fields and not for each sub-field, the velocity of each sub-field is related to the corresponding mixture field by a linear slip relation. We also express the new border velocities by a linear relation with the new mixture liquid velocity:

$$\begin{aligned} u_f^{n+1} &= C_{slip,f}u_k^{n+1} + U_{slip,f} \\ u_b^{n+1} &= C_{0l}u_l^{n+1} + U_0 \end{aligned} \tag{9}$$

In the present study we apply $C_{slip,f} = 1$ and $U_{slip,f} = 0$ (no-slip), meaning that all sub-fields travel with the same velocity as the mixture field.

Since some of the control volumes are moving, sections can both grow and decrease in length. Because of this, certain grid management operations are needed: short sections are merged together with neighboring sections of similar type, and long sections might be split into two shorter sections. Sections might also be converted from one type to another, for instance from bubble to slug, or slug to bubble. The grid management is controlled by a grid coefficient for the bubble sections, named Δx_{bubble}. A bubble section is merged with a neighboring bubble section if the length is smaller than the grid coefficient, and split in two if the section length is larger than twice the grid coefficient. Consequently, the length (Δx) of a bubble section is bounded by the range shown in Eq. (10):

$$\Delta x_{\text{bubble}} \le \Delta x \le 2\cdot \Delta x_{\text{bubble}} \tag{10}$$

Slug sections are never split, but are converted to a bubble section when they become smaller than a minimum length (set to 0.1 m in the presented simulations). Because of these grid management operations, some sections can span multiple pipes that could possibly have different properties. As an example, a single slug control volume around the riser bend might be located partially in the near horizontal part before the riser, and partially in the riser. To account for this, the integration over the momentum equation control volume is split up into a sum of integrals over each different pipe segment. This integral is represented by a summation over the subscript index s in the momentum equation, which is shown below in Eq. (11). For instance, L_j is the length of the entire control volume, while $L_{j,\,s}$ is the length of pipe segment s within the control volume.

$$M_{k,j}^n\frac{u_{k,j}^{n+1} - u_{k,j}^n}{\Delta t} + \frac{M_{k,j}^{n+1} - M_{k,j}^n}{\Delta t}u_{k,j}^{n+1}$$
$$+\left[m_{k,J}^{n+1}A_J^n\left(u_{k,J}^{n+1} - u_{b,J}^{n+1}\right)\hat{u}_{k,J}^{n+1} - m_{k,J-1}^{n+1}A_{J-1}^n\left(u_{k,J-1}^{n+1} - u_{b,J-1}^{n+1}\right)\hat{u}_{k,J-1}^{n+1}\right]$$
$$= -V_{k,j}^n\frac{\left(p_J^{n+1} - p_{J-1}^{n+1}\right)}{L_j^n} + \dot{M}_{k,j,s}^{src,n}u_{k,j}^{src,n}$$
$$-\frac{1}{8}\rho_{k,j}^n\left|u_{k,j}^n\right|u_{k,j}^{n+1}\sum_s L_{j,s}^n\Big(\left(1 - W_{v,j,s}^n\right)\lambda_{k,h,j,s}^n S_{k,h,j,s}^n$$
$$+W_{v,j,s}^n\lambda_{k,v,j,s}^n S_{k,v,j,s}^n\Big)$$
$$-\frac{1}{8}\rho_{g,j}^n\left|u_{k,j}^n - u_{m,j}^n\right|\left(u_{k,j}^{n+1} - u_{m,j}^{n+1}\right)\sum_s L_{j,s}^n\Big(\left(1 - W_{v,j,s}^n\right)\lambda_{i,h,j,s}^n S_{i,h,j,s}^n$$
$$+W_{v,j,s}^n\lambda_{i,v,j,s}^n S_{i,v,j,s}^n\Big)$$
$$-\rho_{k,j}^n g\sum_s V_{k,j,s}^n\cos\theta_{j,s}\frac{h_{k,J,s}^n - h_{k,J-1,s}^n}{L_{j,s}^n} - \rho_{k,j}^n g\sum_s V_{k,j,s}^n\sin\theta_{j,s}$$
$$\pm\phi_{e,j}^n u_{l,j}^{n+1} \mp u_{d,j}^{n+1}\phi_{d,j}^n \tag{11}$$

The terms on the right hand ride represent: Pressure gradient, source/sink terms, wall friction, interfacial friction, hydraulic level gradient, gravity, and entrainment and deposition. θ is the pipe inclination, S_k and S_i are the fluid-wall and interfacial perimeters, and h_k the fluid height. λ_k and λ_i are the fluid-wall and interfacial friction factors, and W is a weighting factor between stratified and annular flow which will be described together with the friction models in the following section. u_k^{src} is the velocity of the mass source or sink. For mass sinks the velocity is set equal to the fluid velocity in the cell the sink is located. ϕ_e and ϕ_d are entrainment and deposition rates, while u_l and u_d are the velocities of the liquid film and droplet field respectively (related to the mixture field velocity by a slip relation as shown in Eq. (9)).

The implicit mass, momentum and pressure equations are solved by the following iterative procedure:

1. Insert the pressure equation for the unknown change in pressure in the momentum equation, and solve for new velocities
2. Back-substitute the new velocities in the pressure equation to calculate the new pressure
3. Calculate new border velocities and move borders to new positions
4. Solve mass equation for change in mass, and update densities and holdup
5. Repeat until the volume fraction error is below specified tolerance, using new values for mass and velocity as a new guess for the non-linear terms in the momentum and pressure equation

Unless the system of equations is solved exactly within machine precision, the volume obtained from the masses and densities will not sum exactly to the physical volume (meaning that the holdups do not sum exactly to 1). This error is called the volume fraction error. This discrepancy is corrected for in the next time step, by adding the current volume error in each grid cell as a source term in the pressure equation. This ensures that the sum of the phase fractions will stay close to 1. In this study the volume fraction error was always ensured to converge below 0.001. The volume fraction error (V_{err}) and pressure equation correction term (Ψ_f) are shown below in Eq. (12).

$$V_{err}^n = \frac{\sum_f \frac{M_{f,i}^n}{\rho_{f,i}^n} - V_i^n}{V_i^n}, \qquad \Psi_{f,i}^{n+1} = \frac{\sum_f \frac{M_{f,i}^n}{\rho_{f,i}^n} - V_i^n}{\Delta t^{n+1}} \tag{12}$$

A.1.1. Friction models

Both the wall friction factor (λ_k) and interfacial friction factor (λ_i) are weighted between the horizontal and vertical friction models, depending on the inclination:

$$\lambda S = W_v(\theta)(\lambda_v S_v) + (1 - W_v(\theta))(\lambda_h S_h) \tag{13}$$

Here S represents the fluid-wall perimeter for the wall friction, and the interfacial perimeter for the interfacial friction. Subscripts v and h represent vertical and horizontal. This vertical weighting factor is defined by:

$$W_v = \sin^2(\theta) \tag{14}$$

For negative inclinations we use $W_v = 0$. The wall and interfacial shear stress is expressed by the Darcy friction factor definition (Moody, 1944):

$$\tau_{k,wall} = \tfrac{1}{8}\lambda_k \rho_k u_k |u_k| \qquad \tau_i = \tfrac{1}{8}\lambda_i \rho_g (u_k - u_m)|u_k - u_m| \tag{15}$$

Here u_m is the velocity of the other mixture field. For the wall friction factor, we use the Haaland and Hagen–Poiseuille friction factors (Haaland, 1983; White, 2011), for turbulent and laminar flow respectively:

$$\lambda_k = \left(-1.8\log_{10}\left[\frac{6.9}{\mathrm{Re}_k} + \left(\frac{\varepsilon/D_{hyd,k}}{3.7}\right)^{1.11}\right]\right)^{-2} \tag{16}$$

$$\lambda_{k,wall} = \frac{64}{\mathrm{Re}_k} \tag{17}$$

Here, ε is the hydraulic roughness, Re_k is the Reynolds number, and $D_{hyd,\,k}$ is the hydraulic diameter:

$$\mathrm{Re}_k = \frac{\rho_k D_{hyd,k} u_k}{\mu_k} \qquad D_{hyd,g} = \frac{4A_g}{S_g + S_i} \qquad D_{hyd,l} = \frac{4A_l}{S_l} \tag{18}$$

where μ_k represents the dynamic fluid viscosity. To achieve continuity between the laminar and turbulent friction factors, we use the maximum value of Eqs. (16) and (17). For the interfacial friction factor in horizontal flow we use the following correlation (Andritsos and Hanratty, 1987):

$$\begin{aligned} \lambda_{i,h} &= \lambda_{g,0} && \alpha_g|u_g| < u_{g,crit} \\ \lambda_{i,h} &= \lambda_{g,0}\left[1 + 15\sqrt{\frac{h_l}{D}}\left(\frac{\alpha_g|u_g|}{u_{g,crit}} - 1\right)\right] && \alpha_g|u_g| \geq u_{g,crit} \\ u_{g,crit} &= 5\sqrt{\frac{\rho_{g,atm}}{\rho_g}} \end{aligned} \tag{19}$$

Here $\rho_{g,\,atm}$ is the gas density at atmospheric pressure, h_l is the liquid height and D the inner pipe diameter. The interfacial friction factor for vertical flow (Moeck and Stachiewicz, 1972) is presented in Eq. (20).

$$\lambda_{i,v} = 0.02\left(1 + 1458\left(\frac{\delta_l}{D}\right)^{1.42}\right) \tag{20}$$

Here, δ_l is the mean film thickness in annular flow:

$$\delta_l = \frac{D}{2}\left(1 - \sqrt{1 - \alpha_l}\right) \tag{21}$$

We limit the relative film thickness in Eq. (20) to be maximum 0.2, which is the maximum experimental value it is defined for.

A.1.2. Entrainment and deposition rates

For entrainment in both horizontal and vertical flow and for deposition in vertical flow we use the correlations presented in Eq. (22) (Govan, 1990). The entrainment correlation for vertical flow has been multiplied by a factor of 2 compared to the original correlation presented by Govan, as this was found to give better predictions of the slugging periods and liquid holdup. The purpose of this is not to tune existing correlations to the presented experimental data, but to illustrate the effect of having a good entrainment model when simulating severe slugging.

For deposition in horizontal flow we use a linear relationship with the deposition constant and droplet field concentration. Like for the wall and interfacial friction, the entrainment rates are weighted between the horizontal and vertical models by the weighting factor in Eq. (14) as an attempt to interpolate the entrainment and deposition rates for inclinations between horizontal and vertical configurations.

$$\begin{aligned} \phi_{e,v} &= LS \cdot 2 \cdot 5.75 \cdot 10^{-5}\left[(\max[\mathrm{Re}_{ol} - \mathrm{Re}_{crit}, 0])^2 \frac{\mu_{ol}^2 \rho_{ol}}{D\sigma_{go}\rho_{gg}^2}\right]^{0.316} \\ &\quad \times \alpha_{gg}\rho_{gg}|u_{gg}| \\ \mathrm{Re}_{ol} &= \frac{\alpha_{ol}\rho_{ol}|u_{ol}|D}{\mu_{ol}} \qquad \mathrm{Re}_{crit} = \exp^{5.8504 + 0.4249\frac{\mu_{gg}}{\mu_{ol}}\sqrt{\frac{\rho_{ol}}{\rho_{gg}}}} \\ \phi_{d,v} &= \begin{cases} LS \cdot 0.18 C_{og}, & \frac{C_{og}}{\rho_{gg}} < 0.3 \\ LS \cdot 0.083\left(\frac{C_{og}}{\rho_{gg}}\right)^{-0.65} C_{og}, & \frac{C_{og}}{\rho_{gg}} > 0.3 \end{cases} \\ \phi_{e,h} &= 0.5\phi_{e,v} \\ \phi_{d,h} &= LS \cdot k_{D,h} C_{og} \end{aligned} \tag{22}$$

Here the subscript *ol* refers to *oil in mixture liquid*, *og* refers to *oil in mixture gas*, while gg refers to *gas in mixture gas.* σ_{go} is the gas-oil interfacial tension, and C_{og} is the droplet field concentration, given by:

$$C_{og} = \frac{M_{og}}{V_g} = \frac{\rho_{og}\alpha_{og}}{\alpha_{og} + \alpha_{gg}} \tag{23}$$

The deposition velocity constant for horizontal flow ($k_{D,\,h}$) is set to 0.1 m/s. Both the entrainment and deposition correlations have been multiplied by the control volume length L and pipe perimeter S to get entrainment and deposition rates in the correct units (converting from mass rate per length and surface area kg/(m^2 s) to kg/s).

A.2. Slug border velocities

The coefficients C_{0l} and U_0 in Eq. (9) which relate the slug border velocity to the mixture liquid velocity are dependent on if the slug border is a Taylor bubble nose, or a slug front. For other borders than slug borders these coefficients are set to 0, meaning no movement.

A.2.1. Bubble-nose criterion

In order to determine whether a slug border is a slug front or a bubble nose, one must apply a bubble turning criterion. In the presented model the criterion is based on the assumption that the Taylor bubble tends to move in the opposite direction of the pressure gradient. This criterion has been shown to predict the bubble turning point quite well in (Johansen and Nydal, 2005; Nydal, 1998). The bubble will thus turn when the gravitational forces balance the friction forces at the bubble nose. This can be expressed as:

$$\frac{0.125\rho_l|u_{crit}|u_{crit}\lambda_l S_l}{A} = |\rho_l g \sin\theta| \tag{24}$$

Here, u_{crit} is the absolute value of the critical velocity for which the bubble turns while g is the gravitational acceleration. In laminar flow this equation has a simple analytical solution, by using the laminar friction factor from Eq. (17):

Table 7
Procedure to determine if a slug border is a bubble nose.

	Slug-bubble (right slug border)	Bubble-slug (left slug border)
Nose criterion	$\lvert u_{crit}\rvert \text{sign}(\theta) \le u_{l,slug}$	$\lvert u_{crit}\rvert \text{sign}(\theta) > u_{l,slug}$

$$u_{crit,lam} = \frac{\rho_l g D^2 \lvert\sin\theta\rvert}{32\mu_l} \tag{25}$$

For turbulent flow however, when using the Haaland equation (Eq. (16)), we get an implicit equation for the critical velocity:

$$u_{crit,tur} + 1.8(2gD\lvert\sin\theta\rvert)^{\frac{1}{2}}\log_{10}\left[\frac{6.9\mu_l}{\rho_l D u_{crit,tur}} + \left(\frac{\varepsilon/D}{3.7}\right)^{1.11}\right] = 0 \tag{26}$$

This equation is easily solved with, for instance, Newton's method. To determine whether the laminar or the turbulent critical velocity should be used, we first calculate the critical laminar velocity, and then calculate the corresponding Reynolds number and friction factor using the critical laminar velocity. We then calculate the turbulent friction factor, using the critical velocity obtained from the laminar theory. If this turbulent friction factor is larger than the laminar friction factor and the Reynolds number is above 300, we assume that the turbulent friction model should be used and go on to solve Eq. (26). If not, we use the critical velocity calculated from the laminar friction model. Finally the correct sign is applied: for positive inclinations the critical velocity is negative, while it is positive for negative inclinations. The procedure shown in Table 7 can then be used to determine if the slug border is a nose.

A border that is not a nose is set to be a slug front. Note that it might happen that both the left and right borders of a slug are fronts, or both might be noses.

A.2.2. Slug front velocity

The velocity of a slug front is calculated from a mass balance across the front, giving the following linear relation with the slug velocity:

$$\begin{aligned} U_{front} &= u_{l,slug} + \frac{A_{l,bubble}}{A_{g,bubble}}\left(u_{l,slug} - u_{l,bubble}\right) \\ &= u_{l,slug}\overbrace{\left(\frac{1}{1-\alpha_{l,bubble}}\right)}^{C_{0l}} + \overbrace{\frac{-u_{l,bubble}\alpha_{l,bubble}}{1-\alpha_{l,bubble}}}^{U_0} \end{aligned} \tag{27}$$

The slug front will absorb liquid both because it is moving and because the liquid in the neighboring bubble is moving. To accommodate for this liquid, the absorbed liquid is distributed on top of the liquid in the neighboring bubble.

Note that Eq. (27) is only used if liquid is going into the slug front. If we have a slug front that is not absorbing liquid it is assumed to be a front moving with the mixture liquid velocity, and we set $C_{0l} = 1$ and $U_0 = 0$.

A.2.3. Bubble nose velocity

A bubble velocity relation for inclined pipes has been proposed (Bendiksen, 1984) of the form:

$$u_b = C_{0l}u_l + U_{0v} + U_{0h} \tag{28}$$

Here U_{0v} and U_{0h} represents the vertical and horizontal components of the drift velocity. A similar model to that of Bendiksen is implemented in the presented model:

$$\begin{aligned} u_b &= C_{0,1}u_l + U_{0,v} + \text{sgn}_h\lvert U_{0,h}\rvert \quad F_r{<}3.5 \\ u_b &= C_{0,2}u_l + U_{0,v} \quad F_r{\ge}3.5 \\ C_{0,1} &= \left(1.05 + 0.15\sin^2(\theta)\right) \\ C_{0,2} &= 1.2 \\ U_{0,v} &= 0.351\sin(\theta)\sqrt{\frac{(\rho_l-\rho_g)}{\rho_l}gD} \\ U_{0,h} &= 0.542\cos(\theta)\sqrt{\frac{(\rho_l-\rho_g)}{\rho_l}gD} \end{aligned} \tag{29}$$

The parameters $C_{0,\,1}$ and $C_{0,\,2}$ in Eq. (29) capture the effect of the velocity profile, in which the fluid velocity at the center of the pipe is greater than the average velocity, making the bubble front propagate faster than the mixture. If the mixture velocity profile is directed in the opposite direction of the bubble nose, a lower value for $C_{0,\,1}$ and $C_{0,\,2}$ is expected. Based on the experimental results from Bendiksen (1984), a value of 0.98 is used in such cases. F_r is the Froude number, which represents the ratio of inertia to gravitational forces:

$$F_r = u_l\sqrt{\frac{\rho_l}{gD(\rho_l-\rho_g)}} \tag{30}$$

The parameter sgn_h in Eq. (29) is -1 if the nose is pointing left, and +1 if the nose is pointing to the right. The Taylor bubble tends to travel along the top of the pipe, instead of being symmetric around the pipe center line. This effect is represented by the first line in Eq. (29), when the Froude number is low. When the velocity is increased, the bubble nose is forced toward the center of the pipe, and the second line in Eq. (29) is used. For simplicity, we use the maximum value of these two relations.

References

Andritsos, N., Hanratty, T.J., 1987. Influence of interfacial waves in stratified gas–liquid flows. AIChE J. 33, 444–454.

Bendiksen, K.H., 1984. An experimental investigation of the motion of long bubbles in inclined tubes. Int. J. Multiphase Flow 10, 467–483.

Benjamin, T.B., 1968. Gravity currents and related phenomena. J. Fluid Mech. 31, 209–248.

Courant, R., Friedrichs, K., Lewy, H., 1928. Über die partiellen Differenzengleichungen der mathematischen Physik. Math. Ann. 100, 32–74.

Courant, R., Isaacson, E., Rees, M., 1952. On the solution of nonlinear hyperbolic differential equations by finite differences. Commun. Pure Appl. Math. 5, 243–255.

Dukler, A.E., Hubbard, M.G., 1975. A model for gas–liquid slug flow in horizontal and near horizontal tubes. Ind. Eng. Chem. Fundam. 14, 337–347.

Dumitrescu, D., 1943. Strömung an Einer Luftblase im Senkrechten rohr. Z. Angew. Math. Mech. 23, 139–149.

Govan, A.H., 1990. Modelling of Vertical Annular and Dispersed Two-Phase Flows, Imperial College, London.

Haaland, S.E., 1983. Simple and explicit formulas for the friction factor in turbulent pipe flow. J. Fluids Eng. 105, 89–90.

Jeyachandra, B.C., Gokcal, B., Al-Sarkhi, A., Sarica, C., Sharma, A., 2012. Drift-velocity closure relationships for slug two-phase high-viscosity oil flow in pipes. SPE J. 17, 593–601.

Johansen, M., Nydal, O.J., 2005. Experiments on large gas bubbles in multiphase gas–oil–water pipe flow, HEFAT 2005. In: Fourth International Conference on Heat Transfer, Fluid Mechanics and Thermodynamic.

Joseph, D.D., 2003. Rise velocity of a spherical cap bubble. J. Fluid Mech. 488, 213–223.

Kjeldby, T., 2013. Lagrangian Three-Phase Slug Tracking Methods. Department of Energy and Process Engineering, NTNU, Trondheim.

Kjølaas, J., De Leebeeck, A., Johansen, S.T., 2013. Simulation of hydrodynamic slug flow using the LedaFlow slug capturing model. In: Eighth International Conference on Multiphase Flow. Jeju, Korea.

Kongsberg Oil and Gas Technologies, 2015. LedaFlow.

Moeck, E.O., Stachiewicz, J.W., 1972. A droplet interchange model for annular-dispersed, two-phase flow. Int. J. Heat Mass Transfer 15, 637–653.

Moody, L.F., 1944. Friction factors for pipe flow. Trans. ASME 66, 671–684.

Nydal, O.J., 1998. Experiments in downwards flow on stability of slug fronts. In: Third International Conference on Multiphase Flow. Lyon, France.

Nydal, O.J., Banerjee, S., 1996. Dynamic slug tracking simulations for gas–liquid flow in pipelines. Chem. Eng. Commun. 141–142, 13–39.

SINTEF, 2015. LedaFlow.

White, F.M., 2011. Fluid Mechanics. McGraw Hill, New York, NY.

Paper 2

Comparison of numerical methods for slug capturing with the two-fluid model

---

Maurice H.W. Hendrix, Ivar Eskerud. Smith, Joost S.B. van Zwieten, Benjamin Sanderse

ICMF-2016 – 9th International Conference on Multiphase Flow
May 22nd–27th 2016, Firenze, Italy.

In this paper the capability of the two-fluid model to describe the transition from stratified to slug flow is investigated, by employing three different numerical discretization techniques: classical finite volume, discontinuous Galerkin, and a Lagrangian finite volume approach (Sluggit). The paper was written together with Ph.D. students from the Delft University of Technology and my co-supervisor Benjamin Sanderse. The Sluggit code was in also used for the additional simulations presented below the paper.

Comparison of numerical methods for slug capturing with the two-fluid model

Maurice H.W. Hendrix[1], Ivar E. Smith[2], Joost S.B. van Zwieten[1], Benjamin Sanderse[3]

[1] *Delft University of Technology, Delft, The Netherlands*

[2] *NTNU, Trondheim, Norway*

[3] *Shell Technology Centre Amsterdam, Amsterdam, The Netherlands*

Abstract

In this paper the capability of the two-fluid model to describe the transition from stratified to slug flow is investigated, by employing three different numerical discretization techniques: classical finite volume, discontinuous Galerkin, and a Lagrangian finite volume approach. It is shown that stratified wavy flow can transition from well-posed to ill-posed following the Kelvin-Helmholtz instability mechanism. In the ill-posed regime grid convergence cannot be obtained. However, with low order discretization methods, or coarse grids and time steps, well-posed numerical solutions can still be obtained. Such solutions should however be critically assessed because they seem to be physical while in fact they are meaningless. The conditional well-posedness of the two-fluid model therefore requires a careful discretization in order to use it for slug capturing.

Keywords: two-fluid model, slug capturing, well-posedness, finite volume method

1. Introduction

In the petroleum industry multiphase flow occurs when transporting oil and gas through long multiphase pipeline systems. The behaviour of the flow can take many forms, depending on parameters like fluid velocities, pipe properties and fluid properties. An important flow regime is slug flow, in which liquid pockets, separated by gas bubbles, propagate in an alternating fashion with high speed along the pipeline. Such slugs have a large influence on the sizing of receiving facilities such as slug catchers or separators. The industry uses various flow models for simulating slug flow, but there is a need for increased accuracy. A promising approach is using so-called slug capturing, through the accurate numerical solution of the one-dimensional two-fluid model. This approach is believed to be capable of describing the transition from stratified flow to slug flow, see e.g. Ref. [1].

One of the issues in the transition from stratified flow to slug flow is that the two-fluid model can become ill-posed, see e.g. Ref. [2] and Ref [3]. Reference [2] mainly focuses on the effect of the spatial discretization and employs an incompressible model. Reference [3] also discusses the incompressible model and performs linear and nonlinear stability analyses. In this paper we instead consider the full compressible model and study several spatial and the temporal discretization methods and on ill-posedness and on convergence and stability. The paper is organized as follows: in section 2 the two-fluid model equations are explained, in section 3 the different discretization methods are presented, and in section 4 results are shown for the Kelvin-Helmholtz instability

2. Governing equations of the two-fluid model

The governing equations of the one-dimensional two-fluid model consist of a mass and momentum conservation equation for each phase:

$$\frac{\partial}{\partial t}\left(\rho_k A_k\right)+\frac{\partial}{\partial x}\left(\rho_k A_k u_k\right)=0 \tag{1}$$

$$\frac{\partial}{\partial t}\left(\rho_k A_k u_k\right)+\frac{\partial}{\partial x}\left(\rho_k A_k u_k^2\right)+A_k\frac{\partial p}{\partial x}+LG_k+\sum_{\substack{\gamma\in\{W,G,L\}\\ \gamma\neq k}}P_{k\gamma}\tau_{k\gamma} \\ +A_k\rho_k g\sin\theta \;=0 \tag{2}$$

supplemented with the constraint $\sum_k A_k = A$. Here A_k is the is cross-sectional area occupied by phase k. The density and the velocity of phase k are denoted by ρ_k and u_k respectively. LG_k represents the hydraulic level gradient term, which can be expressed in conservative form by integrating the hydrostatic pressure over the cross-sectional phase area A_k. The level gradient term for the gas and liquid phase will thus read (van Zwieten et al., Ref [4]):

$$\mathrm{LG}_k=\frac{\partial}{\partial x}\left(\rho_k g\cos\theta(hA_k\pm\frac{1}{12}P_{LG}^3)\right) \tag{3}$$

Note that this model is more complete than the one in Ref. [2] since we take compressibility into account in all terms, including the level gradient term.

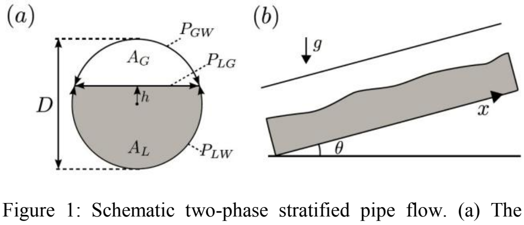


Figure 1: Schematic two-phase stratified pipe flow. (a) The cross-section shows the definition of the perimeters and the interface height h which is defined relative to the pipe center. (b) Cross-section indicating the pipe coordinate x and inclination angle θ.

We assume an isothermal system, so that the density of the gas and liquid phases are given by an equation of state which is a function of pressure only.

2.1. *Friction models*

The wall and interfacial shear stress are expressed by the Fanning friction factor definition:

$$\tau_{k\gamma}=\begin{cases}\dfrac{1}{2}f_k\rho_k u_k\,|\,u_k\,| & \text{if } \gamma=W\\ \dfrac{1}{2}f_{int}\rho_G\left(u_k-u_\gamma\right)|\,u_k-u_\gamma\,| & \text{if } \gamma\in\{G,L\}\end{cases}\tag{4}$$

We model the friction factor f_k of phase k with the pipe wall with Churchill's relation (Ref [5]):

$$\begin{aligned} f_k &= 2\left(\left(\frac{8}{\text{Re}}\right)^{12}+\left(A+B\right)^{-1.5}\right)^{\frac{1}{12}}\\ A &= \left(2.457\ln\left(\left(\left(\frac{7}{\text{Re}_k}\right)^{0.9}+0.27\frac{\varepsilon}{D_{hk}}\right)^{-1}\right)\right)^{16}\\ B &= \left(\frac{37530}{\text{Re}_k}\right)^{16}\end{aligned}\tag{5}$$

Here ε is the hydraulic pipe roughness, Re_k is the Reynolds number,

$$\text{Re}_k=\frac{\rho_k u_k D_{hk}}{\mu_k}\tag{6}$$

and D_{hk} is the hydraulic diameter:

$$D_{hk}=\begin{cases}\dfrac{4A_L}{P_{LW}} & \text{if } k=\text{L}\\ \dfrac{4A_G}{P_{GW}+P_{GL}} & \text{if } k=\text{G}\end{cases}\tag{7}$$

The interfacial friction factor f_{int} is calculated by:

$$f_{int}=\max\{f_G,0.014\}\tag{8}$$

3. Discretization techniques

Three different numerical discretization techniques are investigated in this paper: classical finite volume (CFV), discontinuous Galerkin (DG), and a Lagrangian finite volume (LFV) approach. The discontinuous Galerkin method combines features of both finite element and finite volume methods. The CFV and LFV methods share many similarities, though several aspects like discretization, solution procedure and pressure-velocity coupling are treated differently. The LFV model is also capable of employing moving control volumes, though this aspect of the model is not of primary focus in this study and is thus not described in the LFV model description. All three models employ a staggered grid, illustrated in Figure 2.

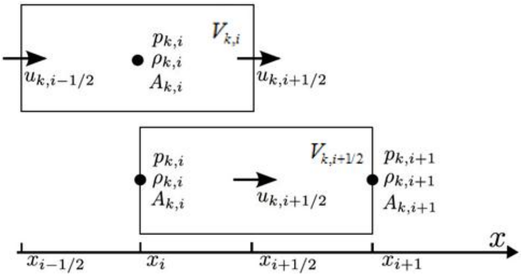


Figure 2: Staggered grid lay-out that is used for the finite volume schemes.

The discrete equations for the mass and momentum equations of the three aforementioned models are presented in the following sections. Super-script n will be used to denote the time index, sub-script i represents the spatial index, while sub-script k represents the gas or liquid phase. A "hat" symbol (^) is placed above unknown new variables where a convection scheme like upwind or central difference is used.

3.1. *Classical Finite Volume (CFV)*

In this section the discretization of the classical finite volume approach will be explained. First the spatial discretization is discussed, followed by the temporal discretization. Finally, the interpolation of unknown quantities is discussed.

Mass conservation equation:

The mass conservation equation is discretized by integrating Eqn (1) over the p-volume $V_{k,i}$. In our 1D framework this results in an integral in the x-direction which yields:

$$\frac{d}{dt}\left(\rho_{k,i}V_{k,i}\right)+\left(\rho A\right)_{k,i+1/2}u_{k,i+1/2}-\left(\rho A\right)_{k,i-1/2}u_{k,i-1/2}=0\tag{9}$$

Momentum conservation equation:

In a similar way we integrate Equation (2) over the u-volume ($V_{k,i+1/2}$) to obtain

$$\begin{aligned}&\frac{d}{dt}\left(\rho_{k,i+1/2}u_{k,i+1/2}V_{k,i+1/2}\right)+\left(\rho A\right)_{k,i+1}(u_{k,i+1})^2-\left(\rho A\right)_{k,i}(u_{k,i})^2\\ &\quad+A_{k,i+1/2}\left(p_{i+1}-p_i\right)+\text{LG}_{k,i+1/2}\\ &\quad+\sum_{\substack{\gamma\in\{\text{G,L,W}\}\\ \gamma\neq k}}P_{k\gamma,i+1/2}\tau_{k\gamma,i+1/2}\Delta x_{i+1/2}\ +\ \rho_{k,i+1/2}\ g\sin\theta\ V_{k,i+1/2}=0\end{aligned}\tag{10}$$

$\text{LG}_{k,i+1/2}$ is the discrete level gradient. For the gas phase the discrete level gradient is calculated as:

$$\text{LG}_{\text{G},i+1/2}=g\cos\theta\left(\rho_{\text{G},i+1}\left(hA_\text{G}+\frac{P_{LG}^3}{12}\right)_{i+1}-\rho_{\text{G},i}\left(hA_\text{G}+\frac{P_{LG}^3}{12}\right)_i\right)\tag{11}$$

The level gradient of the liquid phase is approximated in a similar fashion. For the central scheme, the unknown variables are calculated by:

$$\hat{\phi}_{i,\mathrm{CD}} = \frac{1}{2}\left(\phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}}\right), \quad \hat{\phi}_{i+\frac{1}{2},\mathrm{CD}} = \frac{1}{2}\left(\phi_i + \phi_{i+1}\right) \tag{12}$$

For the FOU scheme, any unknown variable ϕ is taken from the direction the flow is coming. Since the velocities are all positive in the current test case, we get:

$$\hat{\phi}_{i,\mathrm{FOU}} = \phi_{i-\frac{1}{2}}, \quad \hat{\phi}_{i+\frac{1}{2},\mathrm{FOU}} = \phi_i \tag{13}$$

Time integration:

In order to advance the two-fluid model in time, a composite vector $\boldsymbol{U}$ which contains mass and momentum at all grid points is created. If we define $M_{k,i} \equiv \rho_{k,i} V_{k,i}$ and $P_{k,i+1/2} \equiv \rho_{k,i+1/2} u_{k,i+1/2} V_{k,i+1/2}$, this vector, for N grid points, will have the form:

$$\begin{aligned}\boldsymbol{U} = [&M_{(G,1)}, \ldots, M_{(G,N)}, M_{(L,1)}, \ldots, M_{(L,N)}, \\ &P_{(G,1+1/2)}, \ldots, P_{(G,N+1/2)}, P_{(L,1+1/2)}, \ldots P_{(L,N+1/2)}]^T\end{aligned} \tag{14}$$

The complete semi-discrete system can be then written as:

$$\frac{d\boldsymbol{U}}{dt} = F(\boldsymbol{U}) \tag{15}$$

In this formulation we substituted the constraint to close the system. The temporal discretization used for the classical finite volume scheme is a BDF scheme:

$$\frac{1}{\Delta t}\left(a_0 \boldsymbol{U}^{n+1} + a_1 \boldsymbol{U}^n + a_2 \boldsymbol{U}^{n-1}\right) = \beta F(\boldsymbol{U}^{n+1}) \tag{16}$$

We will consider two BDF schemes. The first one is a BDF1 scheme, which is essentially a backward Euler scheme. For BDF1 the coefficients read: $a_0 = 1$, $a_1 = -1$, $a_2 = 0$ and $\beta = 1$. The second scheme we consider is the second order BDF2 scheme with coefficients $a_0 = 1$, $a_1 = -4/3$, $a_2 = 1/3$ and $\beta = 2/3$.

For both the BDF1 and BDF2 scheme Eqn (16) constitutes a nonlinear system that needs to be solved for $\boldsymbol{U}^{n+1}$, which we achieve by using a Newton approach. The fact that the system is solved for $\boldsymbol{U}^{n+1}$, which contains the mass and momentum at each grid point, guarantees mass and momentum conservation independent of time step and grid size.

3.2. Lagrangian Finite Volume (LFV)

The LFV code is also a finite volume method, like the CFV code, but features some distinct differences:

- Possibility to use moving control volumes (not used in the current study).
- The constraint is implemented via a pressure equation.
- The squared velocity in the convective momentum term consists of one central interpolation multiplied by a selected convection scheme, like first order upwind. The CFV model on the other hand uses the squared value of the selected convection scheme.

Mass conservation equation:

$$\frac{M_{k,i}^{n+1} - M_{k,i}^{n}}{\Delta t} = \hat{m}_{k,i-1/2}^{n+1} A_{i-1/2}^{n} u_{k,i-1/2}^{n+1} - \hat{m}_{k,i+1/2}^{n+1} A_{i+1/2}^{n} u_{k,i+1/2}^{n+1} \tag{17}$$

Here, m_k is the specific mass, defined by:

$$m_k = \rho_k \alpha_k = \frac{M_k}{V} \tag{18}$$

while α_k and V are the hold-up fractions and total cell volume respectively.

Momentum conservation equation:

The momentum equation is solved for the change in velocity:

$$\begin{aligned}&M_{k,i+1/2}^{n} \frac{u_{k,i+1/2}^{n+1} - u_{k,i+1/2}^{n}}{\Delta t} + \frac{M_{k,i+1/2}^{n+1} - M_{k,i+1/2}^{n}}{\Delta t} u_{k,i+1/2}^{n+1} \\ &\quad + m_{k,i+1}^{n+1} A_{i+1}^{n} u_{k,i+1}^{n+1} \hat{u}_{k,i+1}^{n+1} - m_{k,i-1}^{n+1} A_{i-1}^{n} u_{k,i-1}^{n+1} \hat{u}_{k,i-1}^{n+1} = \\ &\qquad - V_{k,i+1/2}^{n} \frac{\left(p_{i+1}^{n+1} - p_{i-1}^{n+1}\right)}{\Delta x_{k,i+1/2}^{n}} - \Delta x_{k,i+1/2}^{n} LG_k^n \\ &\qquad - \tau_{kW,i+1/2}^{n} P_{kW,i+1/2}^{n} \Delta x_{k,i+1/2}^{n} - \tau_{GL,i+1/2}^{n} P_{GL,i+1/2}^{n} \Delta x_{k,i+1/2}^{n} \\ &\qquad - M_{k,i+1/2}^{n} g \sin\theta\end{aligned} \tag{19}$$

The level gradient term is discretized identical to what is done in the CFV method.

Time integration and pressure-velocity coupling:

In contrast to the CFV code, the LFV code implements the constraint by deriving a pressure equation. This equation is obtained by expanding the time derivative of mass in the continuous mass equation, Eqn (1), by the product rule, dividing by the fluid density and summing this equation over all phases:

$$\sum_k \frac{V_{k,i}^{n+1}}{\rho_{k,i}^{n}} \left[\left(\frac{\partial \rho_{k,i}^{n}}{\partial p}\right)_T\right] \frac{p_i^{n+1} - p_i^{n}}{\Delta t} = \sum_k \left(\frac{1}{\rho_{k,i}^{n}} \begin{bmatrix} \hat{m}_{k,i-1/2}^{n+1} A_{i-1/2}^{n} u_{k,i-1/2}^{n+1} \\ -\hat{m}_{k,i+1/2}^{n+1} A_{i+1/2}^{n} u_{k,i+1/2}^{n+1} \end{bmatrix}\right) + \Psi_i^n \tag{20}$$

The term Ψ_i^n represents a correction for a possible volume fraction error from the previous time step:

$$\Psi_i^n = \frac{\displaystyle\sum_k \frac{M_{k,i}^{n}}{\rho_{k,i}^{n}} - V_i^n}{\Delta t^{n+1}} \tag{21}$$

The momentum equation (Eqn (19)) first is solved for the change in velocity, using Eqn (20) to eliminate the unknown new pressure directly by substitution. After the momentum equation has been solved, the new velocity is inserted in the pressure equation (Eqn (20)) to obtain the change in pressure. The mass equation is then solved for the change in mass, and the procedure is repeated the volume fraction error (deviation from 1 in the sum of hold-up fractions) drops below 1e-8 in all simulations.

3.3. Discontinuous Galerkin (DG)

The third discretization scheme is the space-time Discontinuous Galerkin Finite Element Method (short DG) described in Ref [4]. A DG scheme is similar to a (continuous) Finite Element scheme with the notable exception that basis functions are discontinuous at element edges. This enables the use of stabilization mechanisms developed for Finite Volume schemes and naturally allows for nonconforming meshes. Due to being part of the family of Finite Element Methods it is relatively easy to construct a high-order scheme by increasing the order of the basis functions.

In this paper we use a third-order, piecewise polynomial basis in both space and time, which gives a fourth-order accurate scheme for linear pde's or non-linear pde's with sufficiently smooth solutions. We use a structured, equidistant mesh with rectangular elements encompassing the complete space-time domain. Due to the structure of the mesh and causality in time, it is not necessary to solve the discrete problem on the complete mesh at once. Instead we separate the mesh in a sequence of time-slabs consisting of all elements with the same time interval and solve the discrete problem per time-slab, starting with the first. Note that this procedure is very similar to the time stepping methods used for the Finite Volume schemes, with the difference that with DG a solution is obtained for an entire time-slab at once while with the Finite Volume schemes a solution is obtained at a single point in time per iteration.

The stabilization method is adaptation of Roe's method: as a reference state for linearization we use the average solution value at the element edges and the eigenvalue problem is solved numerically. The complete non-linear discrete system for one time-slab is solved using Newton's method and the linear subproblem using a sparse, direct solver. For more details we refer the reader to van Zwieten et al. (Ref [4]).

4. Results

4.1. Introduction

The test case we discuss considers the evolution of stratified flow to slug flow according to the Kelvin-Helmholtz instability mechanism, and is the same as described in the study by Liao et al. (Ref [2]) and van Zwieten et al. (Ref [4]). We investigate the effect of the different discretization methods on the growth of an initially smooth wave.

The pipeline and fluid properties are given in Table 1. L is the pipe length, D is the inner pipe diameter, θ is the pipe inclination and ε the pipe roughness.

Table 1: Pipeline and fluid properties.

L	D	θ	ε	ρ_l	μ_g	μ_l
[m]	[m]	[°]	[m]	[kg/m^3]	Pa·s	Pa·s
1	0.078	0	1e-8	1000	1.8e-5	8.9e-4

The liquid phase is assumed to be incompressible with density ρ_l. The density of the gas phase is given by:

$$\rho_G = \frac{p\rho_{ref}}{p_{ref}}, \tag{22}$$

where p_{ref} and ρ_{ref} are 10^5 Pa and 1.1614 kg/m^3 respectively.

The initial condition is a sinusoidal wave with the mean value and amplitude for the primitive variables listed in Table 2. The wave number is $k = 2\pi\ m^{-1}$ and the angular frequency ω is approximately 8.484 s^{-1}. For more information we refer to van Zwieten et al. (Ref [4]).

Table 2: Initial conditions.

$u_{G,\text{mean}}$	$u_{G,\text{amp}}$	$u_{L,\text{mean}}$	$u_{\text{L,amp}}$	$\alpha_{l,\text{mean}}$	$\alpha_{l,\text{amp}}$	P_{mean}	P_{amp}
[m/s]	[m/s]	[m/s]	[m/s]	[-]	[-]	[Pa]	Pa
13.82	0.25	1	7e-3	0.5	0.01	10^5	3.7

The mean values were computed by choosing the gas velocity and liquid holdup, and computing the resulting liquid velocity and pressure gradient from the steady state momentum balance obtained by combining the gas and liquid momentum equations, eliminating the pressure gradient (balancing friction and gravity). These initial values result in a required pressure gradient of 74.23 Pa/m, which was added as driving force (source term) to the momentum equations. Periodic boundary conditions are applied.

By computing the characteristic roots of the system of mass and momentum equations, the following well-posedness criterion can be obtained:

$$\left(u_G - u_L\right)^2 < \left(\frac{\alpha_L}{\rho_L} + \frac{\alpha_G}{\rho_G}\right)\frac{\rho_L - \rho_G}{\dfrac{\partial \alpha_l}{\partial h_l}} g\cos\left(\theta\right) \tag{23}$$

This criterion is identical to the Inviscid Kelvin-Helmholtz criterion (IKH) derived by Barnea and Taitel (Ref [6]), and gives the inviscid limit at which the two-fluid model becomes ill-posed (characteristic roots becomes complex).
According to Eqn (23), the test case is well-posed at the initial conditions specified in Table 2. However, a more detailed eigenvalue analysis shows that the initial condition is in the (viscous) well-posed unstable region, and consequently the initial perturbations will grow.

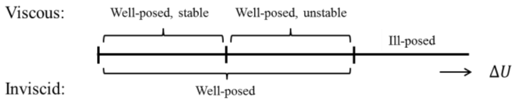


Figure 3: Schematic of stability and well-posedness limits for the two-fluid model.

4.2. Convergence behaviour in well-posed and ill-posed regions

Simulations were run with the different codes, with 40, 80 and 160 grid cells. The time step for each grid is calculated based on the CFL criterion for the liquid velocity: $\Delta t = \text{CFL}_\text{L}\dfrac{\Delta x}{u_L}$. The liquid CFL number (CFL_L) was calculated to be approximately 0.9875, based on the initial wave number and angular frequency, so that we get an integer cycle of the sinusoidal wave after each 60 time steps.

On the left side of Figure 4 the liquid hold-up at different time instances is shown, corresponding to 1, 4 and 7 cycles of the wave moving through the domain. In black the exact analytical solution to the linearized system is added as a reference (note that this is only valid for small times). On the right side two of the four eigenvalues of the two-fluid model are shown (the other two correspond to fast pressure waves associated with acoustics, which are of less importance here). It can be seen that when time increases, the amplitude of the hold-up wave starts to grow. In Figure 4 (a) and (b) we clearly see convergence upon mesh refinement. We also observe that the higher order methods are much more accurate, although a fair comparison requires that we take into account the effect of computational time. In Figure 4 (c) the wave steepens and nonlinear effects are important. It can be seen that in the neighbourhood of the steepening, the real part of the eigenvalues are becoming equal. Closer inspection reveals that the eigenvalues are forming a complex conjugate pair. This indicates that the two-fluid model is not hyperbolic anymore and it therefore becomes ill-posed; Eqn (23) is violated. Related to this is that in the ill-posed region the different discretizations do not converge upon mesh refinement. This means that in essence the results of the two-fluid model have become meaningless. It can be noted that the fourth-order DG scheme already shows ill-posedness for the medium grid N=80, while the other schemes are still well-posed for N=80.

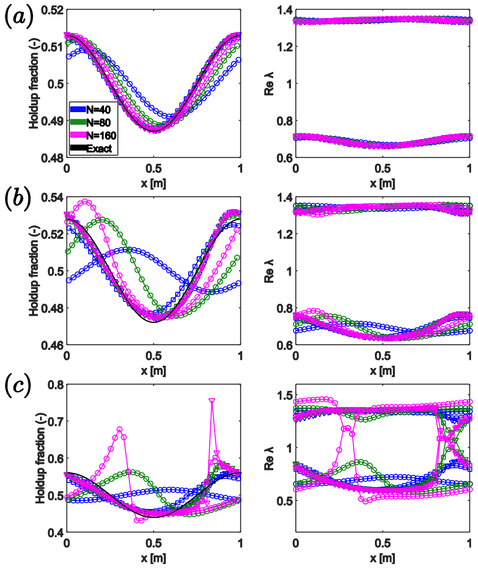


Figure 4: Wave evolution in terms of liquid hold-up and real part of eigenvalues after (a) 1, (b) 4 and (c) 7 cycles. LFV results are depicted with circles (○), CFV (BDF2) results are depicted with squares (□), and DG results are depicted with triangles (▽).

4.3. *Influence of discretization method on predicting ill-posedness*

We further investigate the effect of the discretization on ill-posedness by comparing the time instance at which complex eigenvalues first appear, t_C. For this study we focus on different discretization techniques using solely the CFV scheme. Figure 5 shows that, when the time step goes to zero, all methods converge towards the same t_C. Note that the grid is refined simultaneously with the time step since the CFL number is kept fixed.

When the time step increases, we observe that the lowest order methods start to deviate first. t_C rapidly increases, until a time step is reached for which complex eigenvalues are not found anymore. For time steps larger than this critical time step the simulations are well-posed, even though a refined – and therefore more accurate – simulation would indicate an ill-posed problem.

It is clear that this is an undesirable situation, since one can obtain seemingly meaningful results with a coarse grid or a low order discretization method, that are in fact meaningless. The advantage of higher order methods such as BDF2 instead of BDF1 (Backward Euler) is very clear here: with BDF2 we still obtain the 'correct' ill-posedness at time steps and grid sizes that are around 10 times larger than with BDF1. The advantage of BDF2 is not only apparent in terms of improved accuracy, but also, and maybe more importantly, in capturing the correct mathematical properties of the two-fluid model.

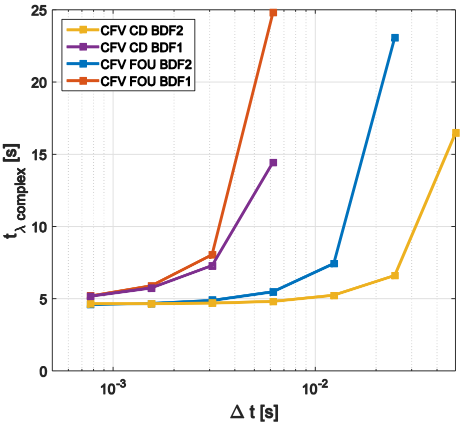


Figure 5: Time instance at which complex eigenvalues first appear, as function of time step, for different discretization methods.

5. Conclusions

In this paper we have investigated the growth of waves in stratified flow as a model for the transition of stratified flow to slug flow in multiphase flow pipelines. In particular, we have studied the effect of different discretization methods on the wave growth and on the onset of ill-posedness.

By studying the behaviour of the eigenvalues of the system of equations in space and time, it appears that during wave steepening the eigenvalues become complex. When simultaneously refining grid and time step, subsequent solutions do not converge, i.e. they do not become grid independent when the eigenvalues are complex. However, given the same initial conditions, well-posed solutions can still be obtained when using low order discretization methods or coarse grids or time steps, although this is undesirable.

The current paper therefore indicates that initially well-posed, but unstable, waves in a stratified flow can grow to become ill-posed, before they have reached the top of the pipe, i.e. before stratified flow has transitioned to slug flow. This indicates that the transition from stratified flow to slug flow, at least for the conditions investigated in this paper, cannot be captured with the two-fluid model, since the model becomes ill-posed. The ill-posedness of the model manifests itself in a lack of convergence upon grid and time step refinement, which essentially renders the simulation results in the ill-posed regime useless. We have shown in this paper that, depending on the discretization method employed, well-posed solutions might still be obtained when using low order discretizations or coarse grids, as typically used in practical studies with commercial simulators. Such solutions should however be critically assessed because they can appear to be physical while in fact they are meaningless.

A number of options are available in literature to circumvent the ill-posedness of the two-fluid model, e.g. the inclusion of surface tension, axial diffusion, a virtual mass force, or a momentum flux parameter. For future work we recommend to investigate the effect of such terms on the transition from stratified to slug flow.

6. References

1. Issa, R.I. and M.H.W. Kempf, *Simulation of slug flow in horizontal and nearly horizontal pipes with the two-fluid model.* International Journal of Multiphase Flow, 2003. **29**(1): p. 69-95.
2. Liao, J., R. Mei, and J.F. Klausner, *A study on the numerical stability of the two-fluid model near ill-posedness.* International Journal of Multiphase Flow, 2008. **34**(11): p. 1067-1087.
3. Fullmer, W.D., V.H. Ransom, and M.A. Lopez de Bertodano, *Linear and nonlinear analysis of an unstable, but well-posed, one-dimensional two-fluid model for two-phase flow based on the inviscid Kelvin–Helmholtz instability.* Nuclear Engineering and Design, 2014. **268**: p. 173-184.
4. van Zwieten, J.S.B., B. Sanderse, M.H.W. Hendrix, C. Vuik, and R.A.W.M. Henkes, *Efficient simulation of one-dimensional two-phase flow with a new high-order Discontinuous Galerkin method*. 2015, Delft University of Technology.
5. Churchill, S., *Friction-factor equation spans all fluid-flow regimes.* Chemical Engineering, 1977. **84**(24): p. 91-92.
6. Barnea, D. and Y. Taitel, *Interfacial and structural stability of separated flow.* International Journal of Multiphase Flow, 1994. **20**: p. 387-414.

Additional simulations presented at ICMF

Some additional simulations that are not included in the conference paper were also presented at the ICMF conference. These simulations are of a 10 km straight and horizontal pipe, with an inner diameter of 0.146 m. and a pressure boundary at the outlet of 10 bar. The temperature remained constant in all simulations. Three flow rate combinations were used, with a gas mass flow rate of 0.2 kg/s in all cases, while the water mass flow rate was 20, 5 and 2.5 kg/s. The steady state solver was used to specify the initial conditions, and after 1000 seconds the gas mass flow rate was increased to 0.4 kg/s over a period of 10 seconds. This creates a holdup wave that will move from the inlet to the outlet. All three flow rate combinations were run with the first order upwind scheme, and the third order NOTABLE scheme, with grid sizes of 128, 256, 512, 1024, 2048, 4096 and 8192 cells. This resulted in a total of 42 simulations. A constant time step was used in all simulations, keeping the CFL number constant (Δt = 10 seconds for grid density 128 cells). Figure 76 shows and example of the holdup wave moving through the pipeline, for the case of a water mass flow rate of 2.5 kg/s. It can be seen that the higher order NOTABLE method gives a sharper front, as expected. As the grid density is increased, the results of the lower order and higher order method converges to a similar result. Similarly to what was shown in the conference paper for the time integration, Figure 77 shows the time it takes until complex eigenvalues occur and the simulation becomes ill-posed. The case with a water mass flow rate of 2.5 kg/s stays well-posed for both schemes at all grid sizes. The case with a water mass flow rate of 5 kg/s becomes ill-posed after the gas rate has been increased, while the case with a water mass rate of 20 kg/s is ill-posed already from the start. In opposite to what is presented in the conference paper, this study focus on the spatial discretization and not the time integration. A similar result is however seen: The lower order methods can give a seemingly well-posed result in cases where the actual result should be ill-posed. It can be seen that using a higher order method will give the correct ill-posed result on a coarser grid than the lower order method. It can also be seen that the lower and higher order methods are converging to the same results at the finest grid size.

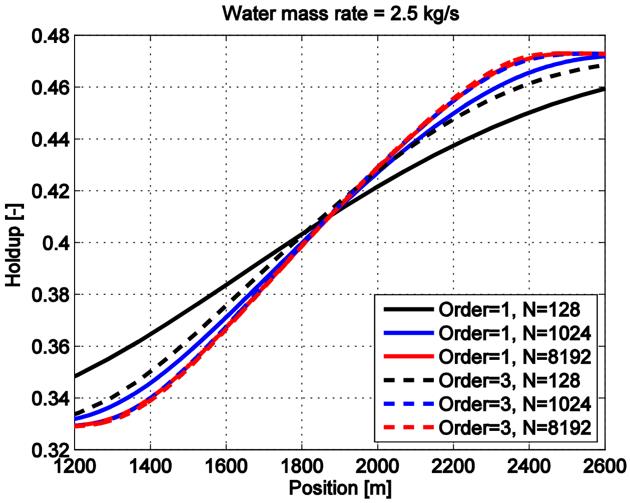


Figure 76: The front of the holdup wave moving through the pipe after the gas mass rate has been increased.

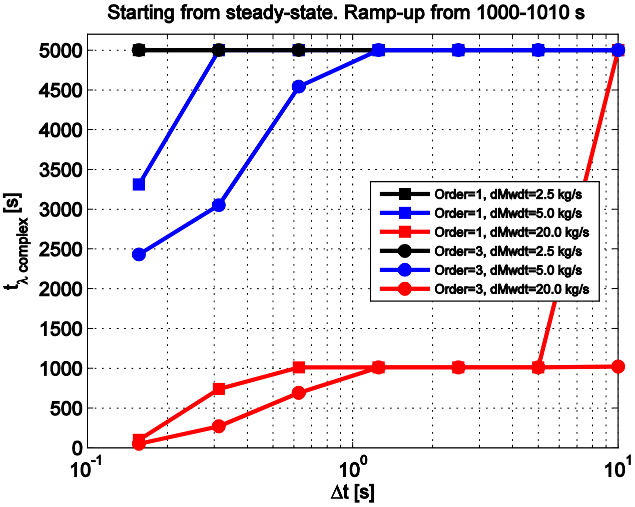


Figure 77: Time until complex eigenvalues occur as function of grid size, for the first and third order upwind schemes. A value of 5000 means that the simulation never becomes ill-posed.

Paper 3

Slug flow simulation, a way to improve air scouring of water mains?

Florent Pourcel, Ivar Eskerud Smith, Sophie Duchesne

18th Conference on Water Distribution System Analysis, WDSA 2016

This paper was written together with a Ph.D. student from the National Institute of Scientific Research (INRS), Québec, Canada. The Sluggit code was used for the simulations, while the experiments were performed at INRS. Advice and discussions for the experimental setup was also given, together with analysing results, creating plots, and writing introduction and results.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 186 (2017) 601 – 608

Procedia Engineering

www.elsevier.com/locate/procedia

XVIII International Conference on Water Distribution Systems Analysis, WDSA 2016

Slug flow simulation, a way to improve air scouring of water mains?

Florent Pourcel[a]*, Ivar Eskerud Smith[b], Sophie Duchesne[a]

[a]*National Institute of Scientific Research, Centre on Water, 490 rue de la Couronne, Québec, G1K 9A9, Québec, Canada*
[b]*Department of Energy and Process Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway*

Abstract

When transporting gas and liquid in the same pipe, waves might become unstable and grow until they bridge the entire pipe cross-section. This flow pattern is called slug flow, and consists of alternating pockets of gas and liquid plugs propagating at high speed through the pipeline. Simulation of slug flow is a topic widely investigated in the oil and gas industry, where slug flow is not desired due to its chaotic nature and increased wall shear stress. The method for cleaning water mains known as air scouring uses this increase of wall shear stress in order to get high efficiency in particle, scale, biofilm and tubercle removal. This is believed to be more effective than unidirectional flushing and to use up to 40 % less water. The efficiency of the cleaning depends on the characteristics of the slugs such as length, velocity and frequency, which are determined by amongst others the water and air superficial velocities, interface friction, fluid properties and pipe properties. The use of slug flow simulation may help to obtain the optimal slug flow characteristics for every sequences of air scouring in order to save both time and water, and ensure the most efficient cleaning of the water mains.

Sluggit is a computational slug flow simulator developed at NTNU for the oil and gas industry. The objective of our work was to validate its predictions for slug flow characteristics within water distribution systems. Slug flow is generated with tap-water and compressed air in a 50 m long PVC pipe with 150 mm of internal diameter. Water distribution system conditions, such as pressure and exit diameter are reproduced. The slug characteristics from the experiments are recorded, and compared with the predictions of Sluggit.

Peer-review under responsibility of the organizing committee of the XVIII International Conference on Water Distribution Systems

Keywords: Slug flow; air scouring; water mains

* Corresponding author. Tel.: +1-418-264-0380; fax:+1-418-654-2600.
E-mail address: florent.pourcel@ete.inrs.ca

Peer-review under responsibility of the organizing committee of the XVIII International Conference on Water Distribution Systems
doi:10.1016/j.proeng.2017.03.276

Nomenclature

D	Pipe diameter [m]
g	Gravitational acceleration [m/s^2]
u_{mix}	Mixture velocity

1. Introduction

Maintaining water distribution systems (WDS) in good condition is necessary for many reasons: keeping water clear and odorless, preventing disinfectant's concentration from lowering rapidly, and avoiding pressure loss because of tubercles or incrustations [1]. Many solutions have been developed through the years such as conventional flushing, air scouring or swabbing, but unidirectional flushing remains the most popular thanks to its low ratio of cost to effectivity [2]. Yet this method also has limitations in efficiency, especially if the network pressure is low, and uses a lot of drinking water [2-4].

Air scouring of water mains is performed in a way close to unidirectional flushing, but compressed air is introduced through a hydrant at the beginning of the section to rinse, as shown in Fig. 1 [2, 4, 5]. This method has some advantages over unidirectional flushing as it is more aggressive (thus more efficient in particles, biofilm, tubercles and incrustations removal due to increased liquid velocity in the slugs and increased wall shear stress), uses as much as 40% less water and doesn't depend on the network pressure. However, it also has disadvantages such as its aggressiveness which doesn't suit to any fragile networks, and the fact that more labor is necessary to close the users' valves during the operation. Because of this reason, which may also imply the need of finding an alternative temporary source of water, air scouring of water mains has a cost which is two to three times higher than unidirectional flushing [2-4, 6].

Air scouring has existed for decades [5], but its operation mainly remains the same nowadays, only based on the manual use of the compressor and the network's valves to generate slug flow within the pipe to be cleaned. Settings are based on charts and judging if slug flow is reached or not is only based on the listening of the water exiting from the outlet hydrant [2, 4, 5].

When gas and liquid are flowing together in the same pipe and high enough velocities, the interfacial friction between them result in waves forming on the interface. Depending on the conditions, these waves might become unstable, and grow large until they bridge the entire pipe cross-section and create a liquid plug (slug) [7]. The gas pressure can then build up behind the liquid slug and accelerate it to high velocities through the pipe. This flow

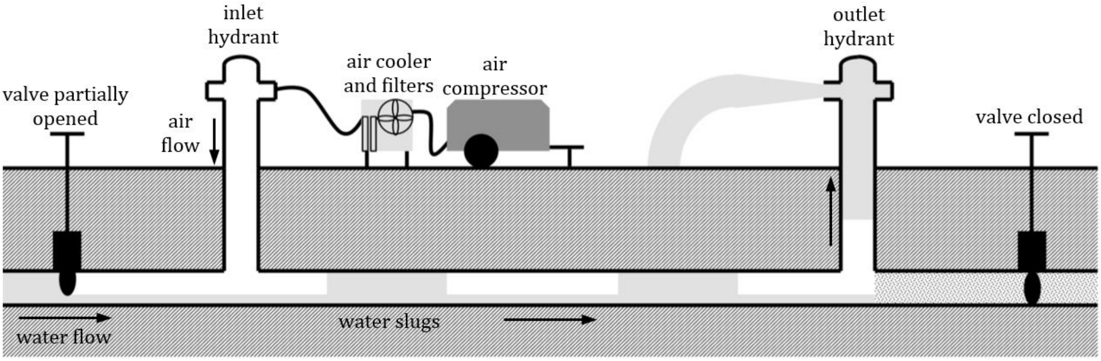


Fig. 1. Illustration of air scouring of water mains.

pattern is called hydrodynamic slug flow (sometimes just referred to as slug flow), and consists of alternating gas regions (Taylor bubbles) and liquid plugs (slugs). The modelling of gas-liquid flow and especially the ability to predict slug flow has been investigated extensively in both the oil and gas industry and in the nuclear reactor industry. The modelling is performed using one-dimensional (1D) averaged conservation equations, as performing a full three dimensional CFD (computational fluid dynamics) simulation of an entire pipeline would take an unreasonable long time and consume a vast amount of computer memory and hard drive space. The 1D averaged equations for mass and momentum conservation of gas-liquid flow are typically referred to as the "two-fluid model". The most commonly used approach is using a coarse grid, together with a sub-grid model which treats slug flow in a statistical manner based on experimental correlations. This type of modelling is often reffered to as the "unit-cell model" approach, based on the concept presented by Dukler and Hubbard [8, 9]. It can only predict the statistical fraction of the pipe covered by slugs (slug fraction) but neither slug length nor slug frequency. Another modelling approach is the so-called slug tracking method [10, 11], which uses Lagrangian tracking of individual slugs and an adaptive grid to avoid numerical diffusion. The slug tracking models are typically also used on a coarse grid and typically use mechanistic initiation criteria to initiate slugs, like the well-posedness criterion [12]. Though slug tracking has been applied a lot in the oil and gas industry, some concerns have been raised with respect to the predictive power of using mechanistic initiation criteria [13]. A relatively new approach for modelling of slug flows referred to as "slug capturing" was first proposed by Issa [14, 15]. In slug capturing the two-fluid model is solved on a fine grid, eliminating the need for a sub-grid model. If the conditions are there for slug flow, it is believed that waves will grow naturally from instabilities and develop into slugs. One of the difficulties with the slug capturing approach is that the resulting slug velocities is a result of both the numerical methods and friction models. Previous work [16] has shown how the interfacial friction models in the commercial oil and gas simulator LedaFlow [17, 18] had to be tuned at high holdup values to obtain slug velocities that corresponds with well-established slug velocity correlations [19]. In addition, there are still some unresolved issues related to the rate of wave growth and grid convergence when refining the grid [20]. The model presented in this study, called Sluggit, is a continuation of the work of [21], and uses a combination of slug capturing and slug tracking: a fine grid is applied for waves to naturally develop and grow into slugs, and the resulting slugs are then tracked using an adaptive grid. A detailed description of the model can be found in [22].

Water, air and pipe characteristics influence the slug velocity, length and frequency. These characteristics can in turn have an impact on the efficiency of the rinsing. This publication describes a part of a wider project dedicated to the comparison of air scouring of water mains versus unidirectional flushing for sedimented and electrostatically adhered particles removal. We aim here to validate the accuracy of Sluggit for the prediction of slug characteristics in a 150 mm inner diameter (ID) water distribution pipe in a lab setup. The objective is to afterwards be able to use simulation tools like Sluggit to predict the optimal air and water settings during real air scouring of water mains operations in order to get the optimal slug flow characteristics, yielding the most efficient removal of particles while saving both water and time.

2. Experimental setup

Experiments were conducted in a PVC loop as shown in Fig. 2. Slugs are formed along a 48 m long (from water injection to outlet 90°) horizontal 150 mm ID PVC pipe having a U-shape due to two long 90° radius bends. Water is pumped from an open tank with a centrifugal pump while air is injected with a rotary screw compressor through a 19 mm ID rubber pipe. Water and air mix within a 100-150 mm ID tee: water comes from the bottom through the vertical 100 mm ID while air comes from a horizontal 150 mm ID, where the 19 mm ID rubber pipe is connected upstream on the top of the PVC pipe. In order to prevent the compressor from introducing oil in the test loop, air is cooled then filtered by a three-stage filtration (coarse, fine and active carbon). The final air temperature is measured in order to have the best possible estimate of the air mass flow rate. Water flow is measured with a magnetic flowmeter (Endress+Hauser Proline Promag 50W, precision ±0.5%) while volumetric air flow is measured by a rotameter (Cole-Parmer Valved Acrylic Flowmeter, 400-3400 LPM, precision ±2%). The end of the loop is made of

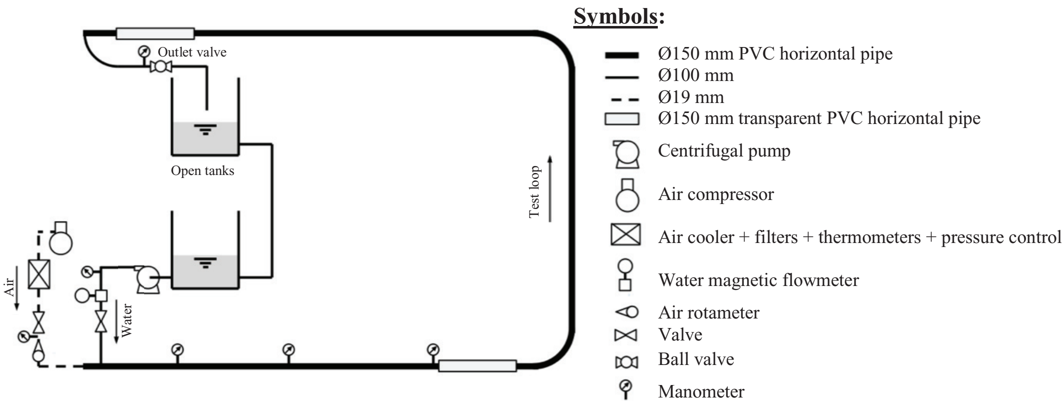


Fig. 2. Illustration of the test loop

a 6 m long 100 mm ID PVC pipe entering vertically down into an open tank directly connected to the first one. To be as close as possible to real network conditions, the outlet valve situated on the 100 mm ID pipe is partially closed in order to simulate the same head loss as obtain from a 100 mm to 63.5 mm reduction in diameter, the latter corresponding to a standard hydrant outlet diameter in Canada [23]. Also, following recommendations of [2] and [5], air pressure right upstream the inlet is set a little less than 0.5 bar below water pressure (see Table 1 for test case conditions). Two test cases, named A and B, were conducted. Note that while performing test case B the compressor was at its maximum capacity. Two 2 m long transparent PVC sections were installed in order to observe the slugs. The first window is situated 20 m downstream the water injection, while the second window is located at 44 m. Slugs are studied by recording 15 minute long videos at 120 images per second with a Sony α7s camera. Two visual marks were created on each transparent section, separated by 1.165 m at the first window and 1.130 m at the second. Slug velocities are calculated by counting the number of images separating the instant their front and the bubble nose reach each mark. Considering previous data and that the highest mean velocity observed was 6.76 m/s, we estimate the precision of this method as being ±5%. Because of the proximity of the second transparent section with the 100 mm ID pipe and the outlet valve, we did not consider bubble nose velocities at this section as the slugs were disturbed by these changes in diameter. Fig. 3 shows the front of a slug as observed at the second transparent section during case A.

Two experiments were conducted at different gas and liquid velocities, and compared with results from the simulations. The test cases are described in Table 1. All experiments were run at a temperature of 10 ºC, and the pressure in the outlet tank was atmospheric. The simulations were performed with mass flow rates as input, calculated from the measured volumetric flow rates, pipe cross sectional area and fluid densities. A constant pressure outlet boundary was used, with atmospheric pressure. Four different grid sizes were tested: 4, 2, 1 and 0.5 times the pipe diameter.

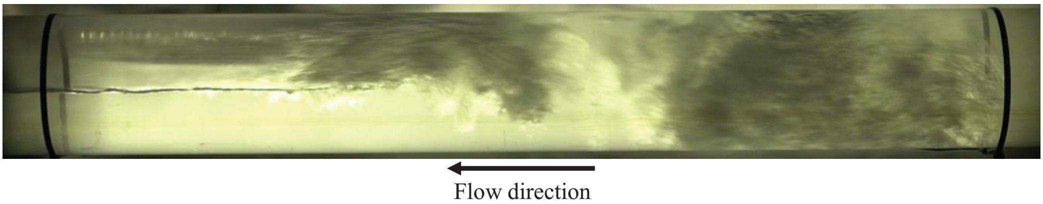


Fig. 3. A slug front formed at the second transparent window in case A.

Table 1. Experimental test cases. The pressures are differential related to atmospheric pressure.

Case name	Air pressure [bar]	Water pressure [bar]	Air mass rate [kg/s]	Water mass rate [kg/s]
A	2.4	3.0	0.051	5.5
B	2.0	3.1	0.092	10.79

The results from the simulations with Sluggit are compared with the experimental measurements in Fig. 4 - Fig. 6. Fig. 4 shows the slug frequency, while Fig. 5 and Fig. 6 show the slug velocity for case A and B, respectively. Appendix A shows the repartition of front and nose velocity during simulation at finest grid and experiment for each case and each position. The theoretical nose velocity included in Fig. 5 and Fig. 6 is calculated from the correlation presented in [19], which for a horizontal pipe reads:

$$u_b = \max\left(1.2u_{mix}, 1.05u_{mix} + 0.542\sqrt{gD}\right) \tag{1.1}$$

Here u_{mix} is the mixture velocity of gas + liquid, while D is the pipe diameter.

2.1. Case A

For the case with the lowest flow rates (case A), it can be seen from Fig. 4 that the slug frequency from the simulations does not compare that well with the slug frequency from the experiments. For the case with the larger flow rate however, the slug frequency is shown to approach the experimental value at both positions in the pipe when refining the grid size. In case A slug flow was observed in the experiments already a few meters after the mixing point of gas and liquid by opening and investigating one of the manometer taps on the pipe. This could mean

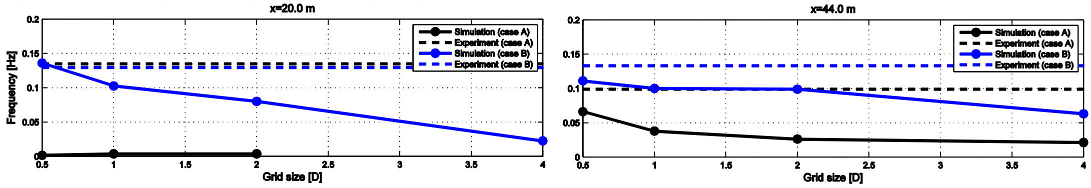


Fig. 4. Slugging frequency from the simulations compared with the experimental values, as function of grid size in units of pipe diameter.

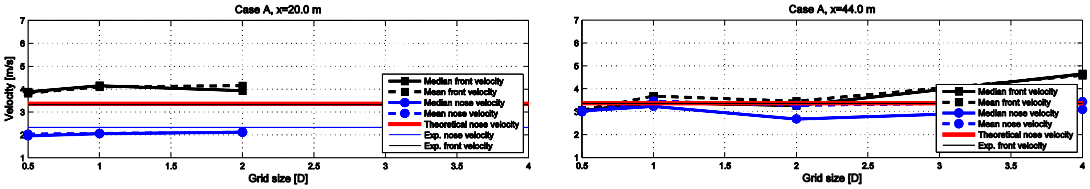


Fig. 5. Comparison of slug velocities between the simulation and experiment for case A as function of grid size in units of pipe diameter.

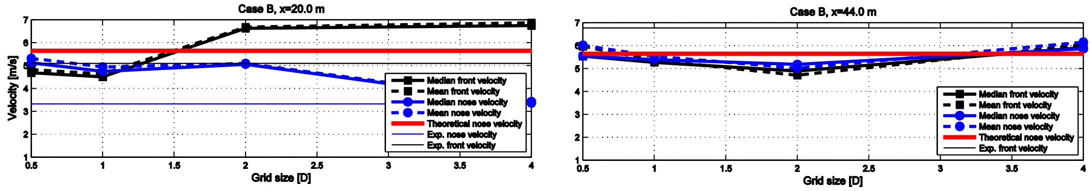


Fig. 6. Comparison of slug velocities between the simulation and experiment for case B as function of grid size in units of pipe diameter.

that slugs are being formed at the inlet, by the configuration of how the gas and liquid are introduced together into the test section. This hypothesis is also supported by the fact that the experimental slug frequency is decreasing from the first to the last transparent window (black dashed line in Fig. 4). The slug frequency from the simulations is however increasing with increasing distance, as more slugs are allowed to develop from the initial stratified flow introduced at the inlet. The two-fluid model is not capable of predicting such inlet generated slug flow, which would probably require fine scale modelling with detailed information about the inlet configuration. However, far into the pipeline the inlet generated slugs should not matter; the flow should then have developed into a steady state flow statistically speaking, with slug frequency and velocities as if the gas and liquid were introduced in a controlled stratified manner. It can also be seen in Fig. 4 that the bubble-nose velocity from the simulations and experiments match relatively well though they are not close to the theoretical value for fully developed slugs. A possible reason for this discrepancy could be that the slugs are still developing and have not reached a steady length and velocity, while the theoretical correlation (equation (1.1)) is only valid for fully developed flow. The fact that the experimental front velocity matches the theoretical nose velocity quite well is most likely a coincidence; this should only be the case when the front and nose velocities are the same, after the slugs have reached a stable length and velocity.

Looking at position 44 m there is a much better correspondence in slug velocity between the experiments and simulations. Note that the experimental nose velocity has been left out from the plot at this location, as the slugs have entered the vertical outlet pipe and are in contact with the outlet valve when the bubble-nose reaches the second window. The theoretical correlation shown in equation (1.1) is only valid for a Taylor bubble in fully developed flow in a single straight pipe and can thus not be used. It might then even happen that the left end of the slug is no longer a bubble-nose, it could also be a slug front if the liquid behind it is moving faster than the slug since the slug is being slowed down by the valve. The slug frequency at 44 m is now closer to the experimental value for case A, but not as large. The reason for this could be that slugs created by the inlet configuration still affects the flow so that a fully developed flow is not yet reached at 44 m. In the simulations however it can be seen that the front and nose velocities are quite similar at the second transparent window, meaning that the slugs in the simulations might have reached a steady length.

2.2. Case B

For the case with larger flow rates there is a quite good match of the slugging frequency, at the finest grid. This is however one of the difficult issues with slug modelling that is yet to be fully understood: the results are dependent on the grid size, and previous studies have shown how the wave growth in the two-fluid model changes with the grid size [20]. The study of [20] also illustrates how several attempts that has been made over the last decades to try to fix this issue, but so far no good solution can be said to exist. Consequently, some testing is required to determine the adequate grid size to use for a specific model implementation.

Looking at the velocities at 20 m, one can see that the simulation nose velocity is not that far from the theoretical value (from equation (1.1)). Furthermore, the front velocity is not that far off the experimental value. The experimental nose velocity is not that close to the simulations or theory, which could be because the slugs are still not fully developed at this location. At location 44 m one can see similar results as at 20 m, but the simulation nose velocity is even closer to the theoretical value and quite close to the front velocity.

3. Conclusions

Experiments with air and water has been presented and compared with simulations from a Lagrangian slug tracking model (Sluggit). The results indicate that the slug frequency obtained from the simulations is strongly dependent on the grid size, while the slug velocity is less sensitive to this. The experiments also indicate that how the gas and liquid are introduced might create slugs directly at the inlet, which can yield a different slug frequency than what one would get from introducing the fluids in a stratified and controlled manner. In the latter case, slugs are being created from growing instabilities, which should be the dominating mechanism for slug flow far into the pipes.

The match in simulated and observed velocity is already encouraging from an air scouring point of view, as shear stress, which depends on the water velocity, is a major factor for sediment removal. Further work is needed in order to get a more accurate match in frequency, as the number of slugs scouring the pipe might also be an important factor.

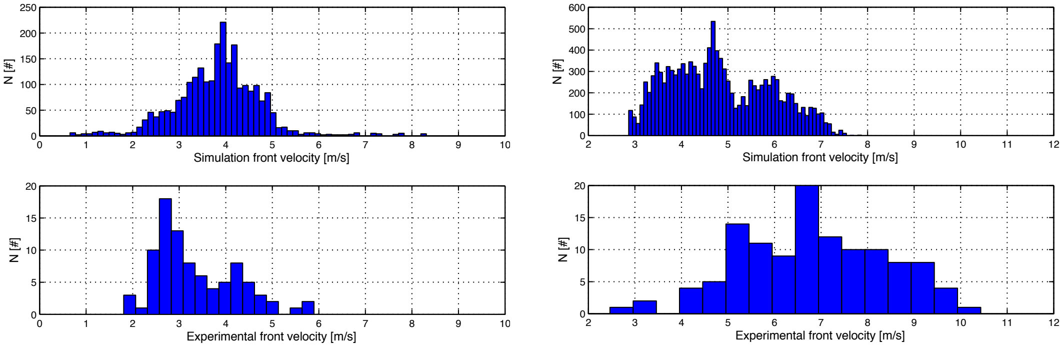
Next steps of the project will be the study of the slugs' characteristics that have the greatest impact on pipe cleaning, and their behaviour in real network.

Acknowledgements

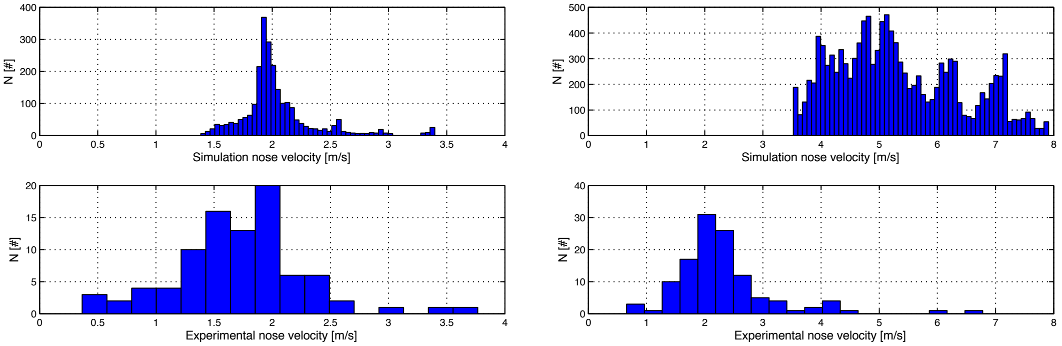
The authors wish to thank the Canadian company NORDIKeau for their agreement in the use of their test loop in order to perform the tests presented in this paper, and Shell Technology Norway for their financial support of the development of the simulation tool (Sluggit).

Appendix A. Repartition of front and nose velocity

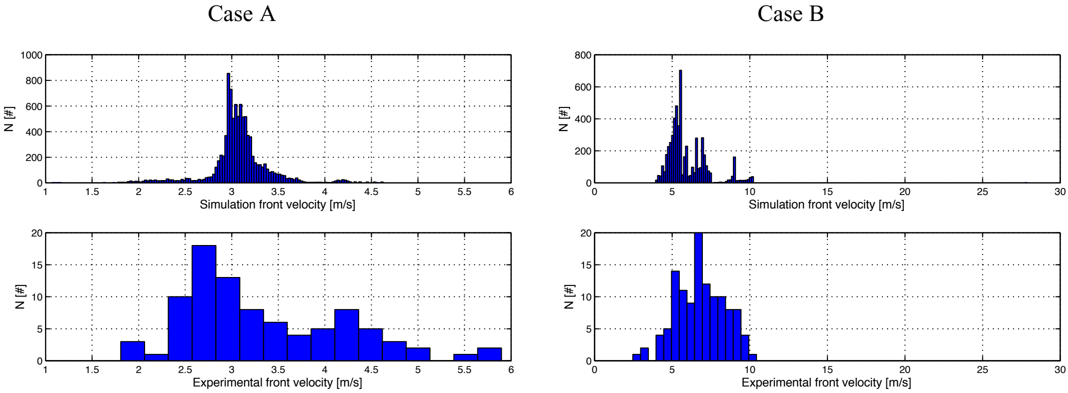
Case A Case B



Front velocity at 20 m



Nose velocity at 20 m



Front velocity at 44 m

References

1. Hasit, Y.J., *Cost and benefit analysis of flushing*. 2004, Denver, Colo.: AWWA Research Foundation : American Water Works Association. xviii, 110 p.
2. Ellison, D. and AWWA Research Foundation., *Investigation of pipe cleaning methods*. 2003, Denver, CO: AWWA Research Foundation and American Water Works Association. xix, 142 p.
3. Kitney, P., R. Woulfe, and S. Codd, *AIR SCOURING OF WATER MAINS - AN ASSET MANAGEMENT APPROACH*, in *64th Annual Water Industry Engineers and Operators' Conference*. 2001: All Seasons International Hotel - Bendigo. p. 48-56.
4. Vitanage, D., F. Pamminger, and T. Vourtsanis, *Maintenance and survey of distribution systems.* Safe Piped Water: Managing Microbial Water Quality in Piped Distribution Systems, 2004: p. 70-85.
5. Elvidge, A.F., *Air Scouring of Water Mains - A Method of Operation*. 1982, Water Research Centre.
6. Shore, D.G. and B.C. Lythell, *Practical experience of water-mains renovation in four UK water companies*. 1992, Water Supply. p. 131-142.
7. Guo, B., et al., *Preface*, in *Offshore Pipelines (Second Edition)*. 2014, Gulf Professional Publishing: Boston. p. xiii-xiv.
8. Dukler, A.E. and M.G. Hubbard, *A Model for Gas-Liquid Slug Flow in Horizontal and Near Horizontal Tubes.* Industrial & Engineering Chemistry Fundamentals, 1975. **14**(4): p. 337-347.
9. Eskerud Smith, I., J. Nossen, and T. Erling Unander, *Improved Holdup and Pressure Drop Predictions for Multiphase Flow with Gas and High Viscosity Oil*. BHR Group.
10. Larsen, M., et al., *PeTra - A Novel Computer Code for Simulation of Slug Flow.* SPE, 1997.
11. Nydal, O.J. and S. Banerjee, *Dynamic slug tracking simulations for gas-liquid flow in pipelines.* Chemical Engineering Communications, 1996. **141-142**(1): p. 13-39.
12. Barnea, D. and Y. Taitel, *Kelvin-Helmholtz stability criteria for stratified flow: viscous versus non-viscous (inviscid) approaches.* International Journal of Multiphase Flow, 1993. **19**(4): p. 639-649.
13. Burke, N.E. and S.F. Kashou, *Slug-Sizing/Slug-Volume Prediction: State of the Art Review and Simulation.*
14. Issa, R., *Simulation of intermittent flow in multiphase oil and gas pipelines*, in *Seventh International Conference on CFD in the Minerals and Process Industries*. 2009: Melbourne, Australia.
15. Issa, R.I. and M.H.W. Kempf, *Simulation of slug flow in horizontal and nearly horizontal pipes with the two-fluid model.* International Journal of Multiphase Flow, 2003. **29**(1): p. 69-95.
16. Kjølaas, J., A. De Leebeeck, and S.T. Johansen, *Simulation of Hydrodynamic Slug Flow Using the LedaFlow Slug Capturing model*. BHR Group.
17. Kongsberg Oil and Gas Technologies. *LedaFlow*. 2015; Available from: http://www.kongsberg.com/en/kogt/products%20and%20services/flow%20assurance/.
18. SINTEF. *LedaFlow*. 2015; Available from: http://www.sintef.no/sintef-materialer-og-kjemi/programvare-2/ledaflow/.
19. Bendiksen, K.H., *An experimental investigation of the motion of long bubbles in inclined tubes.* International Journal of Multiphase Flow, 1984. **10**(4): p. 467-483.
20. Fullmer, W.D., V.H. Ransom, and M.A. Lopez de Bertodano, *Linear and nonlinear analysis of an unstable, but well-posed, one-dimensional two-fluid model for two-phase flow based on the inviscid Kelvin–Helmholtz instability.* Nuclear Engineering and Design, 2014. **268**: p. 173-184.
21. Kjeldby, T., *Lagrangian Three-phase Slug Tracking Methods*, in *Department of Energy and Process Engineering*. 2013, NTNU: Trondheim.
22. Eskerud Smith, I. and O.J. Nydal, *The effect of boundary conditions and droplet entrainment on severe slugging using a Lagrangian slug tracking model.* 2016: International Journal of Multiphase Flow (Revision 2 of manuscript submitted for publication).
23. NFPA. *National Fire Protection Agency*.

Paper 4

Analysis of time integration methods for the compressible two-fluid model for pipe flow simulations

Benjamin Sanderse, Ivar Eskerud Smith, Maurice H.W. Hendrix

International Journal of Multiphase Flow
Volume 95, October 2017, Pages 155-174

This paper was written together Ph.D. student Maurice Hendrix from the Delft University of Technology, and my co-supervisor Benjamin Sanderse. Many different simulations were run for this study, using both my own Matlab code and a code from Shell. We all did simulations, validating each other's results. A von Neumann analysis and a dispersion analysis were also conducted both by myself and by the team from Shell, double-checking each other's work. In the end, we had too much material for one paper. Some of the material I worked on that was discarded from the paper is instead presented below the paper in this thesis.

International Journal of Multiphase Flow 95 (2017) 155–174

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow

Analysis of time integration methods for the compressible two-fluid model for pipe flow simulations

Benjamin Sanderse[a,b,*], Ivar Eskerud Smith[c], Maurice H.W. Hendrix[a,d]

[a] Shell Technology Centre Amsterdam, Amsterdam, The Netherlands
[b] Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands
[c] Norwegian University of Science and Technology (NTNU), Trondheim, Norway
[d] Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 9 March 2017
Revised 9 May 2017
Accepted 15 May 2017
Available online 25 May 2017

Keywords:
Two-fluid model
Time integration method
BDF2
Discrete flow pattern map
Stability
Von Neumann analysis

ABSTRACT

In this paper we analyse different time integration methods for the two-fluid model and propose the BDF2 method as the preferred choice to simulate transient compressible multiphase flow in pipelines. Compared to the prevailing Backward Euler method, the BDF2 scheme has a significantly better accuracy (second order) while retaining the important property of unconditional linear stability (*A*-stability). In addition, it is capable of damping unresolved frequencies such as acoustic waves present in the compressible model (*L*-stability), opposite to the commonly used Crank–Nicolson method. The stability properties of the two-fluid model and of several discretizations in space and time have been investigated by eigenvalue analysis of the continuous equations, of the semi-discrete equations, and of the fully discrete equations. A method for performing an automatic von Neumann stability analysis is proposed that obtains the growth rate of the discretization methods without requiring symbolic manipulations and that can be applied without detailed knowledge of the source code.

The strong performance of BDF2 is illustrated via several test cases related to the Kelvin–Helmholtz instability. A novel concept called Discrete Flow Pattern Map (DFPM) is introduced which describes the *effective* well-posed unstable flow regime as determined by the discretization method. Backward Euler introduces so much numerical diffusion that the theoretically well-posed unstable regime becomes numerically stable (at practical grid and timestep resolution). BDF2 accurately identifies the stability boundary, and reveals that in the nonlinear regime ill-posedness can occur when starting from well-posed unstable solutions. The well-posed unstable regime obtained in nonlinear simulations is therefore in practice much smaller than the theoretical one, which might severely limit the application of the two-fluid model for simulating the transition from stratified flow to slug flow. This should be taken very seriously into account when interpreting results from any slug-capturing simulations.

1. Introduction

In the petroleum industry multiphase flow occurs when transporting oil and gas through long multiphase pipeline systems. The behaviour of the flow can take many forms, depending on parameters like fluid velocities, pipe properties and fluid properties. An important flow regime is (hydrodynamic) slug flow, in which liquid pockets, separated by gas bubbles, propagate in an alternating fashion with high speed along the pipeline. Such slugs have a large influence on the sizing of receiving facilities such as required supports of the pipeline system and on the sizing of the downstream facilities such as separators (or slug catchers), compressors, pumps, and heat exchangers. The industry uses various flow models for simulating slug flow, but there is a need for increased accuracy. A promising approach is using so-called slug capturing, through the accurate numerical solution of the one-dimensional two-fluid model. Although this two-fluid model can become ill-posed, this approach is believed to be capable of describing the transition from stratified flow to slug flow, see e.g. Issa and Kempf (2003). The use of the compressible two-fluid model is necessary to generate slugs at the right frequency (Issa and Kempf, 2003).

Since the slug capturing approach is based on accurate solutions of the underlying model, higher order (higher than 1) numerical schemes or fine meshes are typically employed in both space and time. Different spatial discretization schemes have been

[*] Corresponding author at: Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands.
E-mail addresses: b.sanderse@cwi.nl (B. Sanderse), ivar.eskerud@gmail.com (I.E. Smith), m.h.w.hendrix@tudelft.nl (M.H.W. Hendrix).

http://dx.doi.org/10.1016/j.ijmultiphaseflow.2017.05.002
0301-9322/© 2017 Elsevier Ltd. All rights reserved.

used for the two-fluid model, for example the second-order central scheme (Liao et al., 2008), the QUICK scheme, a Roe scheme (Akselsen, 2016), a pseudo-spectral scheme (Holmås, 2010), and a Discontinuous Galerkin method (van Zwieten et al., 2015). The stability and accuracy properties of several of these schemes have been investigated recently by Fullmer et al. (2014); Liao et al. (2008); López de Bertodano et al. (2017) and Akselsen (2016). In all cases the incompressible model is discussed.

For the time integration of the two-fluid model, the workhorse in both industrial and academic codes has been the first order Backward Euler method, due to its stability and damping properties (Bendiksen et al., 1991; Danielson et al., 2005; Issa and Kempf, 2003; Montini, 2011). The large numerical diffusion, however, requires small time steps; the CFL number based on the liquid velocity is not on the order of 1 but has to be much smaller (Issa and Kempf, 2003; Liao et al., 2008). Only a few sources mention higher order time integration methods for the two-fluid model. Kjølaas et al. (2013) mention that the slug-capturing module in LedaFlow (a commercial pipe flow simulator) uses the second order Crank–Nicolson method. Zou et al. (2015) employ the BDF2 (Backward Differentiation Formula) method for the two-fluid model to simulate the water faucet problem and found that it was superior to the first order Backward Euler method. However, a temporal accuracy study was only performed for a linear advection case, and for the other test cases the 'basic' two-fluid model (without level gradient terms) was used, which is unconditionally ill-posed. Also the commercial nuclear safety code RELAP-7 (Berry et al., 2015) advises BDF2 as the time integration method. However, almost no details are given on the stability, accuracy and damping properties of the numerical time integration method.

The purpose of this paper is to analyse different time integration methods for the compressible two-fluid model in terms of accuracy, stability and damping properties, and to use them to construct so-called discrete flow pattern maps. In particular, we show that BDF2 is preferred over Backward Euler and Crank–Nicolson because it combines second order accuracy with *L*-stability (filtering of acoustic waves at large time steps).

The paper is organized as follows: first, in Section 2 the two-fluid model is explained in terms of eigenvalues, stability and flow pattern maps. Section 3 describes the spatial (central, upwind) and temporal (Backward Euler, Crank–Nicolson, BDF2) discretizations used in this work. In Section 4 the stability and accuracy of the time integration methods are discussed by computing the eigenvalues of the spatial discretization and the stability domain of the time integration methods. Section 5 presents a von Neumann analysis applied to the fully discrete problem and a novel method to assess the stability without requiring symbolic manipulations. Section 6 presents three test cases: linear, nonlinear and acoustic wave growth based on Kelvin–Helmholtz instabilities. Section 7 provides concluding remarks.

2. Governing equations and characteristics

2.1. Compressible two-fluid model

The two-fluid model can be derived by considering mass and momentum balances for the stratified flow of oil and gas in a pipeline. The major assumption in the derivation is that of one-dimensional, stratified flow, with the transverse hydrostatic pressure variation introduced via level gradient terms. Furthermore, we assume isothermal flow so that no energy equation is required, and surface tension is neglected. In contrast to the incompressible model, which is commonly discussed in literature, we allow the gas phase to be compressible (the extension to compressible liquid is straightforward). This leads to the presence of acoustic waves in the solution, which in turn has an effect on the choice of time integration method, as will become clear later. With these assumptions, the two fluid model consists of the conservation equations for mass and momentum for the gas and liquid phase, reading:

$$\frac{\partial}{\partial t}(\rho_g A_g) + \frac{\partial}{\partial s}(\rho_g u_g A_g) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho_l A_l) + \frac{\partial}{\partial s}(\rho_l u_l A_l) = 0 \tag{2}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho_g u_g A_g) &+ \frac{\partial}{\partial s}(\rho_g u_g^2 A_g)\\ &= -\frac{\partial p}{\partial s}A_g + \mathrm{LG}_g - \tau_{gl}P_{gl} - \tau_g P_g - \rho_g A_g g_s + F_{\mathrm{body}}A_g,\end{aligned} \tag{3}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho_l u_l A_l) &+ \frac{\partial}{\partial s}(\rho_l u_l^2 A_l)\\ &= -\frac{\partial p}{\partial s}A_l + \mathrm{LG}_l + \tau_{gl}P_{gl} - \tau_l P_l - \rho_l A_l g_s + F_{\mathrm{body}}A_l,\end{aligned} \tag{4}$$

supplemented with the volume equation:

$$A_g + A_l = A. \tag{5}$$

s is the spatial coordinate along the centerline of the pipe. The driving pressure force $F_{\mathrm{body}} = -\frac{\mathrm{d}p_{\mathrm{body}}}{\mathrm{d}s}$ in the gas and liquid momentum equations is required for the simulations that involve periodic boundary conditions. The friction models are described in Appendix A.2. The level gradient (LG) terms differ from the incompressible case and were derived and shown by us in van Zwieten et al. (2015); a similar form was presented in Masella et al. (1998):

$$\mathrm{LG}_g = \frac{\partial \mathrm{HG}_g}{\partial s}, \quad \mathrm{HG}_g = \rho_g g_n \left[(R-h)A_g + \frac{1}{12}P_{gl}^3\right], \tag{6}$$

$$\mathrm{LG}_l = \frac{\partial \mathrm{HG}_l}{\partial s}, \quad \mathrm{HG}_l = \rho_l g_n \left[(R-h)A_l - \frac{1}{12}P_{gl}^3\right]. \tag{7}$$

In Eqs. (1)–(7), the subscript β denotes the phase, either gas ($\beta = g$) or liquid ($\beta = l$). ρ_β denotes the density of phase βA_β the cross-sectional area occupied by phase β, R the pipe radius, h the height of the liquid layer measured from the bottom of the pipe, u_β the phase velocity, p the pressure at the interface, τ_β the shear stress (with the wall or at the interface), g the gravitational constant, φ the local inclination of the pipeline with respect to the horizontal, $g_n = g\cos\varphi$ and $g_s = g\sin\varphi$. A_l (or A_g) and h are related by a non-linear algebraic expression since the pipeline has a circular cross-section (for channel flow one simply has $h = A_l$). Similarly, the wetted and interfacial perimeters P_g, P_l and P_{gl} can be expressed in terms of the hold-up or the interface height (see Appendix A.1 for more details). As a result, the two-fluid model features five equations with five unknowns (A_g, A_l, u_g, u_l, p). Depending on the velocity difference between the phases, the two-fluid model is well-posed or ill-posed (Barnea and Taitel, 1994; Lyczkowski et al., 1978; Stewart and Wendroff, 1984), as will be discussed in the next section.

2.2. Characteristics

The governing equations of the two-fluid model can be written in quasi-linear form in terms of the primitive variables $\boldsymbol{W} \in \mathbb{R}^q$ ($q = 4$),

$$\boldsymbol{W} = \begin{pmatrix} A_l \\ u_l \\ u_g \\ p \end{pmatrix}, \tag{8}$$

reading

$$\boldsymbol{A}(\boldsymbol{W})\frac{\partial \boldsymbol{W}}{\partial t} + \boldsymbol{B}(\boldsymbol{W})\frac{\partial \boldsymbol{W}}{\partial s} + \boldsymbol{C}(\boldsymbol{W}) = \boldsymbol{0}. \tag{9}$$

The eigenvalues can be found by substituting wave-like solutions in the homogeneous part of the equations (Drew and Passman, 1998; Hirsch, 1994), i.e. $\boldsymbol{W} = \hat{\boldsymbol{W}} e^{I(n_s s + n_t t)}$, leading to

$$\det(n_t \boldsymbol{A} + n_s \boldsymbol{B}) = 0. \tag{10}$$

By letting $\lambda = -n_t/n_s$ $(=\frac{\mathrm{d}s}{\mathrm{d}t})$ this can be seen as the generalized eigenvalue problem

$$\det(\boldsymbol{B} - \lambda \boldsymbol{A}) = 0. \tag{11}$$

Compact expressions for the eigenvalues are difficult to obtain analytically since the characteristic equation is quartic. Evje and Flåtten (2003) derived approximate expressions using a perturbation technique suggested by Toumi and Kumbaro (1996), but did not include the level gradient terms (while their inclusion is essential for well-posedness). When considering an incompressible liquid and a compressible gas, while neglecting the hydraulic gradient term in the gas phase and using the 'incompressible' form of the hydraulic gradient of the liquid phase, a simple expression for the characteristic equation can be obtained (see also Figueiredo et al., 2017):

$$A_g(c_g^2 - (\lambda - u_g)^2)\left(\frac{\partial \mathrm{HG}_l}{\partial A_l} + \rho_l(\lambda - u_l)^2\right) + A_l c_g^2 \rho_g (\lambda - u_g)^2 = 0. \tag{12}$$

Here c_g is the speed of sound of the gas phase ($\rho_g = p/c_g^2$). In our simulations we continuously check the well-posedness of the model by computing the eigenvalues from a quartic which is similar to (12), but which includes all compressibility and hydraulic gradient effects. When complex eigenvalues are encountered, the simulation is stopped, since we take the view of Drew and Passman (1998) that the original initial boundary value problem has become meaningless. The four eigenvalues are typically ordered as follows:

$$\lambda_1 \le \lambda_2 \le \lambda_3 \le \lambda_4, \tag{13}$$

where λ_1 and λ_4 are related to the speed of sound of the gas, and λ_2 and λ_3 are related to the convective wave speeds. The problem contains therefore two distinctly different time scales:

$$|\lambda_1|, |\lambda_4| \gg |\lambda_2|, |\lambda_3|, \tag{14}$$

with a typical ratio of $\mathcal{O}(10^2)$ - $\mathcal{O}(10^3)$. For slightly compressible flow, the eigenvalues λ_2 and λ_3 are expected to be close to the incompressible expressions, see for example (Bonizzi, 2003). These incompressible roots are given by

$$\lambda_{2,3} = \frac{(\rho u)^* \pm \xi}{\rho^*}, \tag{15}$$

where, in the notation from Akselsen (2016):

$$\zeta^* = \frac{\zeta_l}{A_l} + \frac{\zeta_g}{A_g}, \tag{16}$$

and

$$\xi = \sqrt{\rho^*(\rho_l - \rho_g) g_n \frac{\partial h}{\partial \alpha_l} - \frac{\rho_g \rho_l}{A_g A_l}(u_g - u_l)^2}. \tag{17}$$

The incompressible two-fluid model is therefore well-posed as long as the velocity difference between the phases is smaller than the 'inviscid Kelvin–Helmholtz' limit (IKH) (Liao et al., 2008):

$$(u_g - u_l)^2 \le \Delta U_{\mathrm{IKH}}^2 = \frac{(\rho_l - \rho_g)}{\rho^*} g_n \frac{\partial h}{\partial \alpha_l}. \tag{18}$$

The compressibility of the gas hardly changes this limit (Bonizzi, 2003). The compressible two-fluid model under consideration is therefore conditionally well-posed. Several remedies have been proposed to make it unconditionally well-posed:

- An artificial interfacial pressure term (Evje and Flåtten, 2003); this is a mathematical 'trick' which eliminates the growth of interfacial waves due to Kelvin–Helmholtz instabilities (Fullmer et al., 2014; Liao et al., 2008).
- Surface tension (Ramshaw and Trapp, 1978); next to well-posed the model is also stable for small scale perturbations below a cut-off wavelength. However, between this cut-off wavelength and the IKH wavelength the model is unstable (Fullmer et al., 2014) and when sufficiently fine grids are used in numerical simulations, these short-wavelength instabilities are resolved and will grow (Holmås, 2008).
- Diffusive terms, either artificial (Holmås, 2008) or physical (Fullmer et al., 2014). Although this has certain advantages, such as nonlinear stability, the growth rate for short waves is very large, making the model practically ill-posed (Fullmer et al., 2014).

Other approaches are the use of a virtual mass force (Montini, 2011), a momentum flux parameter (Song, 2003; Song and Ishii, 2001) and a two-pressure model (Fitt, 1989; Ransom and Hicks, 1984; 1988). The 'best' remedy is still the subject of debate in the two-fluid model community. In this paper the focus is on the accurate time-integration of the conditionally well-posed model (1)–(4) and we do not make attempts to make the model unconditionally well-posed.

2.3. Stability

To investigate the stability of the equations of the two-fluid model we take the following approach (Montini, 2011; Ramshaw and Trapp, 1978): linearization of the governing equations around a reference state, assuming a travelling wave solution for the perturbations, and determining the dispersion relation to identify the stability. A similar approach is taken in Liao et al. (2008) and Fullmer et al. (2014).

We linearize the governing equations by writing $\boldsymbol{W} = \boldsymbol{W}_0 + \tilde{\boldsymbol{W}}$ (where $\tilde{\boldsymbol{W}} \ll \boldsymbol{W}_0$), using the fact that the reference state (indicated by $(.)_0$) satisfies the equations, and neglecting products of disturbances. The linearized equations then read:

$$\boldsymbol{A}_0 \frac{\partial \tilde{\boldsymbol{W}}}{\partial t} + \boldsymbol{B}_0 \frac{\partial \tilde{\boldsymbol{W}}}{\partial s} = \boldsymbol{C}_0 \tilde{\boldsymbol{W}}, \tag{19}$$

where $\boldsymbol{A}_0 = \boldsymbol{A}(\boldsymbol{W}_0)$, $\boldsymbol{B}_0 = \boldsymbol{B}(\boldsymbol{W}_0)$, and $\boldsymbol{C}_0 = \left(\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{W}}\right)_0$. It is cumbersome to determine $\boldsymbol{C}_0$ analytically due to the presence of the friction factors, see Eq. (A.7), and in general we determine it numerically.

The perturbation is assumed to be of a wave like form: $\tilde{\boldsymbol{W}} = \boldsymbol{\varepsilon} e^{I(\omega t - ks)}$, with ω being the angular frequency and k the wave number. This leads to

$$(\boldsymbol{A}_0 \cdot (I\omega) - \boldsymbol{B}_0 \cdot (Ik) - \boldsymbol{C}_0)\boldsymbol{\varepsilon} e^{I(\omega t - ks)} = 0. \tag{20}$$

For non-trivial solutions we need

$$\det(\boldsymbol{A}_0 \cdot (I\omega) - \boldsymbol{B}_0 \cdot (Ik) - \boldsymbol{C}_0) = 0. \tag{21}$$

This yields the dispersion relation $\omega(k)$. In the absence of source terms ($\boldsymbol{C}_0 = 0$), the condition for stability ($\mathrm{Im}(\omega) = 0$) is equivalent to the IKH well-posedness limit (18) obtained from the eigenvalue analysis. The dispersion analysis presented here is based on the full set of conservation equations and the resulting KH instability is therefore a so-called dynamic instability (Lopez-de-Bertodano et al., 2013). When source terms are included, the stability condition differs from the well-posedness criterion and the stability

Table 1
Parameter values used in the example of the Kelvin–Helmholtz problem.

Parameter	Value	Unit
α_l	0.5	[-]
u_g	13.815	m/s
u_l	1	m/s
ρ_l	1000	kg/m^3
R	0.039	m
p_0	10^5	N/m^2
c_g	293.43	m/s
g	9.8	m/s^2
μ_g	$1.8 \cdot 10^{-5}$	Pa s
μ_l	$8.9 \cdot 10^{-4}$	Pa s
ϵ	10^{-8}	m
F_{body}	74.225	Pa/m

limit is commonly known as the viscous Kelvin–Helmholtz (VKH) stability limit. This 'viscous' instability is similar to the kinematic waves in shallow water theory (Whitham, 1974) and the continuity waves of Wallis (1969).

2.4. Flow pattern map

An example of the IKH and VKH stability boundaries will be given here for the compressible equations in terms of a flow pattern map. We consider the Kelvin–Helmholtz instability problem, at the same conditions as proposed and analyzed by Liao et al. (2008). A horizontal pipe with a length of one meter is taken ($\varphi = 0$) and its diameter is 78 mm. The density of the gas is given by a perfect gas relation,

$$\rho_g = \frac{p}{c_g^2}, \tag{22}$$

where c_g is taken such that for $p = p_0$ we have $\rho_g = 1.1614$ kg/m^3. The density of the liquid is constant (incompressible). The superficial liquid velocity is set at 0.5 m/s, and the superficial gas velocity at $u_{sg} = 6.908$ m/s. The liquid hold-up, gas velocity, liquid velocity and pressure gradient follow from the steady state momentum balance and are given, together with other parameters, in Table 1. Note that ϵ is the hydraulic wall roughness, and μ is the fluid viscosity.

Liao et al. (2008) report that the stability limit (18) for the incompressible model is $\Delta U_{\text{IKH}} = 16.0768$ m/s. In the compressible model the stability limit slightly changes to $\Delta U_{\text{IKH}} = 16.0355$ m/s. Since at the current conditions $\Delta U = 12.815$ m/s, the model is well-posed and stable. Note that we have employed Biberg's approximate relation for $\alpha_l(h)$, see Eq. (A.5), which leads to a small difference in the stability limit compared to the nonlinear relation.

We consider a single wave with $k = 2\pi$ on a domain $s \in [0, 1]$ m. The dispersion analysis, Eq. (20), then leads to 4 waves with the following eigenvalues and angular frequencies:

$$\boldsymbol{\lambda} = \begin{pmatrix} -279.80 \\ 0.69 \\ 1.34 \\ 307.40 \end{pmatrix} m/s, \qquad \boldsymbol{\omega} = \begin{pmatrix} -1758.05 \\ 4.27 \\ 8.48 \\ 1931.47 \end{pmatrix} + \begin{pmatrix} 4.51I \\ 0.59I \\ -0.35I \\ 4.71I \end{pmatrix} 1/s. \tag{23}$$

The fact that all eigenvalues are real indicates that the initial condition is indeed well-posed, with $\lambda_{1,4}$ close to the speed of sound of the gas and $\lambda_{2,3}$ close to the liquid velocity. The angular frequencies indicate that out of the four waves there is one unstable mode, ω_3, which will grow in time.

When repeating this analysis for different superficial liquid and gas velocities the neutral IKH and VKH stability boundaries can be constructed according to $\text{Im}(\omega) = 0$ - see Fig. 1. These neutral stability boundaries are independent of the wave number k (Barnea and Taitel, 1994; Prosperetti and Tryggvason, 2007). In between the inviscid and viscous stability curves we have a well-posed, unstable solution of the two-fluid model, in which transition from stratified flow to slug flow can possibly occur. The conditions given in Table 1 are indicated by 'current conditions' and they are in this well-posed, unstable regime. Lines of constant hold-up are indicated by dashes in Fig. 1. The resulting flow pattern map and lines are similar to those of Barnea and Taitel (1993). In Fig. 2 the dispersion relation associated to the current conditions is shown (square symbol), together with the dispersion relation for a point in the well-posed stable regime (round symbol) and a point in the ill-posed regime (triangle). On the left (Fig. 2a) ω is plotted as a function of wavelength λ, in a similar way as in Fullmer et al. (2014). This confirms that for short waves ($\lambda \to 0$) the ill-posed model possesses unbounded growth, whereas the well-posed unstable model has a bounded growth rate ($\text{Im}(\omega_3) = -0.35$/s). In all cases the most unstable ω is shown, which corresponds to ω_3 (for the stable case this is the least stable mode). On the right (Fig. 2b) the same plot is shown, but now as a function of k, similar to Liao et al. (2008). This latter form of the dispersion relation will be used in our numerical results.

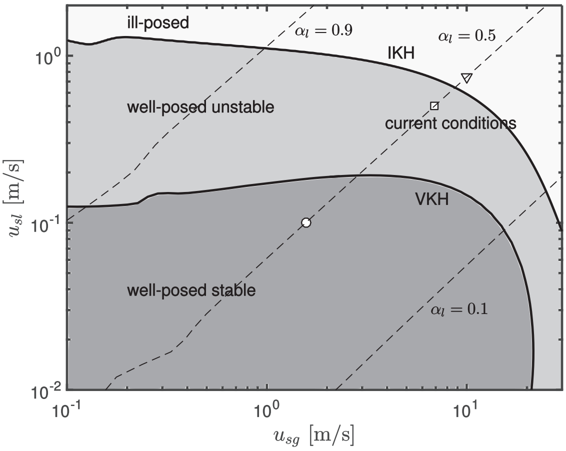


Fig. 1. Flow pattern map based on Kelvin–Helmholtz instabilities. The symbols ○, □ and ▽ correspond to the lines in Fig. 2.

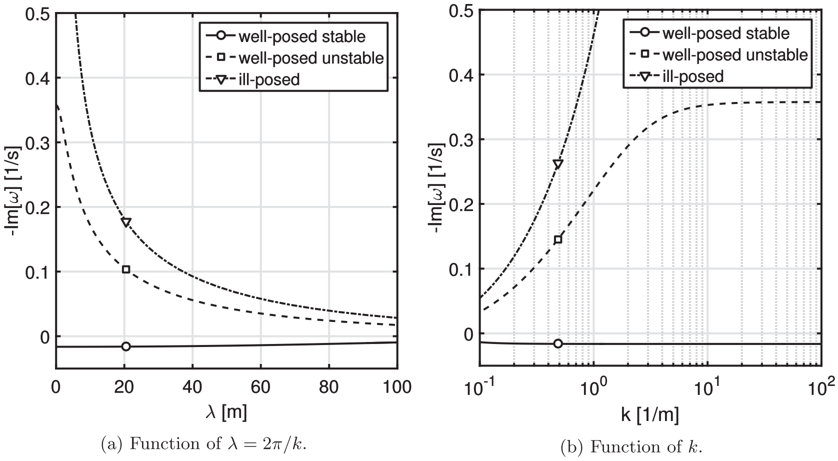
3. Spatial and temporal discretization

3.1. Finite volume method on a staggered grid

As mentioned in the introduction, many options are available for the spatial discretization of the two-fluid model, such as finite difference methods (Fullmer et al., 2014; Liao et al., 2008), pseudo-spectral methods (Holmås, 2010), and characteristics methods (Akselsen, 2016). We discretize the two-fluid model, Eqs. (1)–(4), by using a finite volume method on a staggered grid. As indicated in Fig. 3, the staggered grid consists of both p-volumes, Ω^p, and u-volumes, Ω^u. Each volume consists of a liquid and a gas phase: $\Omega = \Omega_l \cup \Omega_g$, for both u- and p-volumes.

We start with conservation of mass for a phase β (β is liquid or gas). Integration of Eq. (1) in s-direction over a p-volume gives:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\rho_{\beta,i}\Omega_{\beta,i}\right) + \left(\rho_\beta A_\beta\right)_{i+1/2} u_{\beta,i+1/2} - \left(\rho_\beta A_\beta\right)_{\beta,i-1/2} u_{\beta,i-1/2} = 0, \tag{24}$$



(a) Function of $\lambda = 2\pi/k$.

(b) Function of k.

Fig. 2. Dispersion relation for well-posed stable, well-posed unstable, and ill-posed model, corresponding to the points shown in Fig. 1.

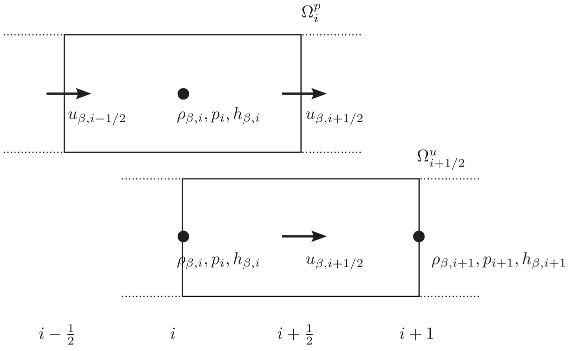


Fig. 3. Staggered grid layout.

with the finite volume size given by

$$\Omega_{\beta,i} = A_{\beta,i}\Delta s_i. \tag{25}$$

The term $(\rho_\beta A_\beta)_{i+1/2}$ requires interpolation from neighbouring values, which is described below. For conservation of momentum we proceed in a similar way. Integration of (3) in s-direction over a u-volume gives:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\rho_{\beta,i+1/2}u_{\beta,i+1/2}\Omega_{\beta,i+1/2}\right) &+ \left(\rho_\beta A_\beta\right)_{i+1}(u_{\beta,i+1})^2 \\ -\left(\rho_\beta A_\beta\right)_i(u_{\beta,i})^2 &= -A_{\beta,i+1/2}(p_{i+1}-p_i) + \mathrm{LG}_{\beta,i+1/2} \\ -\rho_{\beta,i+1/2}\Omega_{\beta,i+1/2}g_s &- \sum_{\substack{\gamma\in\{\mathrm{L,G,W}\}\\ \gamma\neq\beta}} \tau_{\beta\gamma,i+1/2}P_{\beta\gamma,i+1/2}\Delta s_{i+1/2} \\ +A_{\beta,i+1/2}F_{\mathrm{body}}\Delta s_{i+1/2}, &\end{aligned} \tag{26}$$

where

$$\Omega_{\beta,i+1/2} = A_{\beta,i+1/2}\Delta s_{i+1/2}, \tag{27}$$

and the level gradient terms for the gas and liquid are given by (+ for gas, − for liquid)

$$\mathrm{LG}_{\beta,i+1/2} = g\cos(\varphi)\left(\left(hA_\beta \pm \frac{1}{12}P_{gl}^3\right)_i - \left(hA_\beta \pm \frac{1}{12}P_{gl}^3\right)_{i-1}\right). \tag{28}$$

Several terms in (24) and (26) require approximation. All terms that are not part of the convective terms are interpolated using a central scheme, e.g. $A_{\beta,i+1/2} = \frac{1}{2}(A_{\beta,i} + A_{\beta,i+1})$. The convective terms, on the other hand, require more care in order to prevent numerical oscillations. Since the system under consideration is (conditionally) hyperbolic, the wave directions have to be taken into account in the differencing scheme, e.g. by using a Roe method (Akselsen, 2016; Morin et al., 2013). In this work the focus is on the time integration method and we employ standard spatial discretization methods for the convective quantities: first order upwind or second order central. The central scheme is accurate and stable in our computations since the solutions to the test cases are sufficiently smooth, i.e. we are not simulating discontinuities or flow transitions.

The spatial discretization presented in this section leads to a semi-discrete system, only depending on time, which can be written as

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{F}(\boldsymbol{U}), \tag{29}$$

where $\boldsymbol{U} = \boldsymbol{U}(t) \in \mathbb{R}^{qN}$ is the vector of conserved variables. For periodic boundary conditions, which we will consider, it reads

$$\begin{aligned}\boldsymbol{U} = \big[&(\rho_g A_g \Delta s)_{1,\dots,N}, (\rho_l A_l \Delta s)_{1,\dots,N}, (\rho_g A_g u_g \Delta s)_{1/2,\dots,N-1/2}, \\ &(\rho_l A_l u_l \Delta s)_{1/2,\dots,N-1/2}\big]^T.\end{aligned} \tag{30}$$

We use $\boldsymbol{U}$ instead of $\boldsymbol{W}$ in a discrete setting to ensure discrete conservation properties.

3.2. Temporal discretization

Eq. (29) forms a system of ordinary differential equations which can be integrated in time with many different time integration methods (Butcher, 2003; Hairer et al., 2008). A common distinction is between multi-stage methods (Runge–Kutta) and multi-step methods (Adams, Backward Differentiation Formula (BDF)). Within both classes there is a distinction between explicit and implicit methods. Due to the strong time step restriction for explicit methods (caused by the acoustic speeds λ_1 and λ_4), we consider implicit methods. For the two-fluid model, the Backward Euler method is widely applied because of its unconditional numerical stability characteristics, although it has the disadvantage of being only first-order accurate, which introduces a significant amount of numerical diffusion. To construct a second order method that keeps the good stability properties, one can see Backward Euler as a member of the family of implicit Runge–Kutta methods, as a member of the Adams–Moulton methods, or as member of the BDF methods. A second order extension of Backward Euler can be

Table 2
Parameter values for time integration methods.

scheme	a_0	a_1	a_2	θ
Backward Euler	1	-1	0	1
BDF2	$\frac{3}{2}$	-2	$\frac{1}{2}$	1
Crank–Nicolson	1	-1	0	$\frac{1}{2}$

sought in each of these families. We restrict ourselves to Crank–Nicolson and BDF2, which are both second order accurate and unconditionally stable. Backward Euler, Crank–Nicolson and BDF2 can all be gathered under the following expression:

$$\frac{a_0\boldsymbol{U}^{n+1}+a_1\boldsymbol{U}^{n}+a_2\boldsymbol{U}^{n-1}}{\Delta t}=\theta\boldsymbol{F}(\boldsymbol{U}^{n+1},t^{n+1})+(1-\theta)\boldsymbol{F}(\boldsymbol{U}^{n},t^{n}). \tag{31}$$

with the parameter values given in Table 2. The local truncation error for the methods is given by

$$\tau_{\text{BE,CN}}=\left(\theta-\frac{1}{2}\right)\Delta t^2\boldsymbol{F}''+\left(\frac{1}{2}\theta-\frac{1}{3}\right)\Delta t^3\boldsymbol{F}'''+\mathcal{O}(\Delta t^4), \tag{32}$$

$$\tau_{\text{BDF2}}=-\frac{2}{9}\Delta t^3\boldsymbol{F}'''+\mathcal{O}(\Delta t^4). \tag{33}$$

The leading error constant of Crank–Nicolson is $\frac{1}{12}$, which is smaller than that of BDF2.

Eq. (31) is a system of nonlinear equations and application of Newton's method leads to the following linear system that is solved at each time step:

$$\left[\frac{a_0}{\Delta t}\boldsymbol{I}-\theta\left(\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{U}}\right)^m\right]\Delta\boldsymbol{U}=-\left[\frac{a_0\boldsymbol{U}^m+a_1\boldsymbol{U}^n+a_2\boldsymbol{U}^{n-1}}{\Delta t}-\theta\boldsymbol{F}(\boldsymbol{U}^m,t^{n+1})-(1-\theta)\boldsymbol{F}(\boldsymbol{U}^n,t^n)\right], \tag{34}$$

where m is the iteration counter. If the direct evaluation of $\boldsymbol{F}$ in terms of $\boldsymbol{U}$ is not possible - for example in the case of a nonlinear equation of state, when the pressure cannot easily be obtained from the density - we solve Eq. (34) in terms of $\boldsymbol{W}$, which requires the evaluation of an additional Jacobian. Solving system (34) has roughly the same computational cost for Backward Euler, Crank–Nicolson and BDF2. This is because the evaluation of the Jacobian is the most expensive part, which is the same for all methods since they are all implicit in time. From an implementation point of view, BDF2 can be implemented relatively easily in an existing code that uses Backward Euler, because the only extra required variable is $\boldsymbol{U}^{n-1}$.

The BDF2 method suffers from a start-up problem: $\boldsymbol{U}^{n-1}$ is not available at the first time step, which is therefore computed with Backward Euler. Note that for variable time steps the coefficients of the BDF2 method become dependent on the time step ratio. Another issue of BDF2 is the fact that it is not unconditionally monotone (neither is Crank–Nicolson), and therefore under- or overshoots can appear near discontinuities (Hundsdorfer and Verwer, 2007; Zou et al., 2015). Adaptive time stepping or locally using Backward Euler are possible solutions to this issue. In the test cases reported here this was not required.

3.3. Reference solution with explicit Runge–Kutta method

A highly accurate reference solution in time is obtained by employing an explicit Runge–Kutta method. Explicit methods are normally not of interest for the compressible equations due to the time step restriction associated with the acoustic waves, but in the case that these waves are to be resolved (as we will do in one of the test cases), they provide a cheap and highly accurate alternative to implicit methods. A general s-stage explicit Runge–Kutta method applied to (29) reads:

Table 3
Case definition for analysis of stiffness of semi-discrete equations.

Case	N	u_{sl}, u_{sg} [m/s]	c_g [m/s]	Convective terms
I	40	0.5, 6.908	$c_{g,\,0}$	Central
IIa	20	0.5, 6.908	$c_{g,\,0}$	Central
IIb	80	0.5, 6.908	$c_{g,\,0}$	Central
IIIa	40	0.5, 6.908	0.1 $c_{g,\,0}$	Central
IIIb	40	0.5, 6.908	10 $c_{g,\,0}$	Central
IV	40	0.5, 6.908	$c_{g,\,0}$	Upwind
V	40	0.05, 0.829	$c_{g,\,0}$	Central

$$\boldsymbol{U}^{n+1}=\boldsymbol{U}^n+\Delta t\sum_{i=1}^{s}b_i\boldsymbol{F}(\boldsymbol{U}_i,t_i), \tag{35}$$

where

$$\boldsymbol{U}_i=\boldsymbol{U}^n+\Delta t\sum_{j=1}^{s}a_{ij}\boldsymbol{F}(\boldsymbol{U}_j,t_j),\qquad t_i=t^n+c_i\Delta t. \tag{36}$$

We have used a three-stage, third order strong-stability preserving Runge–Kutta method (Gottlieb et al., 2001) with the following non-zero coefficients: $a_{21}=1$, $a_{31}=a_{32}=\frac{1}{4}$, $b_1=b_2=\frac{1}{6}$, $b_3=\frac{2}{3}$.

4. Time scales, stability and stiffness

4.1. Spectrum of the spatial discretization

The distinct time scales of the problem, see Eq. (14), make the semi-discrete Eqs. (29) stiff: some terms in the differential equations lead to very fast transients, while others are much slower. In order to investigate the stiffness induced by the problem and the spatial discretization (without including the effect of the temporal discretization yet) we look at the eigenvalues of the semi-discrete equations. Similar to the stability analysis of Section 2.3, we linearize the semi-discrete equations around a background state (for example the initial condition): $\boldsymbol{U}(t)=\boldsymbol{U}_0+\tilde{\boldsymbol{U}}(t)$, which gives

$$\frac{\mathrm{d}\tilde{\boldsymbol{U}}}{\mathrm{d}t}=\boldsymbol{J}\tilde{\boldsymbol{U}}, \tag{37}$$

with $\boldsymbol{J}=\left(\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{U}}\right)_0\in\mathbb{R}^{qN\times qN}$. Diagonalization of $\boldsymbol{J}=\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{-1}$ leads to the following set of decoupled equations:

$$\frac{\mathrm{d}\tilde{\boldsymbol{Z}}}{\mathrm{d}t}=\boldsymbol{D}\tilde{\boldsymbol{Z}}, \tag{38}$$

where $\boldsymbol{Z}=\boldsymbol{K}^{-1}\boldsymbol{U}$. The matrix $\boldsymbol{D}$ is a diagonal matrix with the eigenvalues μ of $\boldsymbol{J}$. These eigenvalues μ are the discrete approximation of the eigenvalues of only the spatial derivatives of the two-fluid model equations, based on Fourier analysis. For example, a convective term of the form $c\frac{\partial}{\partial s}$ leads to purely imaginary eigenvalues of the form $-cI$. Diffusive phenomena on the other hand are associated to negative real eigenvalues, and unstable solutions to Re$(\mu)>0$. Therefore, in contrast to the eigenvalues λ of the continuous equations, complex eigenvalues μ of the semi-discrete equations are not associated to well- or ill-posedness.

We investigate the behaviour of the eigenvalues μ of the spatial discretization by performing parametric studies on the well-posed, unstable Kelvin–Helmholtz problem for which the conditions where given in Table 1. The considered cases are specified in Table 3. Case I in Table 3 with $N=40$ finite volumes and a central discretization of the convective terms is considered as the base case. Four additional cases are defined, in which the effect of the grid size, gas compressibility, upwind scheme and VKH stability on

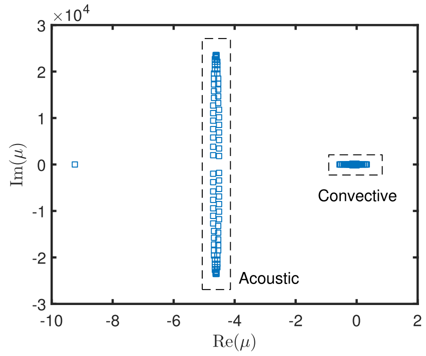


Fig. 4. Eigenvalues for case I.

the eigenvalues is investigated. The eigenvalues $\mu(\boldsymbol{J})$ are computed numerically with Matlab for all cases.

The $4N$ (=160) eigenvalues for case I are shown in Fig. 4, with zoomed-in details for the $\approx 2N$ acoustic modes and $\approx 2N$ convective modes in Figs. 5a and 5 b, respectively. Some convective modes have a positive real part, indicating that certain solution components will grow in time. The central discretization thus captures the fact that the initial condition is in the well-posed unstable regime. The real part of the acoustic eigenvalues corresponds to the imaginary part of the angular frequencies ω_1 and ω_4 (see Eq. (23)). Similarly, the real part of the convective eigenvalues corresponds to the imaginary part of ω_2 and ω_3. The imaginary parts of the acoustic and convective eigenvalues differ by several orders of magnitude. This corresponds to the fact that the acoustic modes have a much higher frequency than the convective modes. The semi-discrete equation system is therefore stiff and requires a suitable time integration method, as will be detailed in Section 4.

The stiffness of the problem depends on the grid size. This is shown in Fig. 5, which compares the eigenvalues for 20 grid points (6 a) and for 80 grid points (6 b). The finer the grid, the larger the imaginary part of the eigenvalues, and the higher the frequencies that can be represented numerically. The scaling of the imaginary part of the eigenvalues is linear in the number of grid points, and when an explicit time integration method is used this leads to a CFL-type stability condition: $\Delta t = \mathcal{O}(\Delta s)$. The fact that the real part of the eigenvalues is not affected by the grid size confirms that there is no numerical diffusion added when the central scheme for the convective terms is employed.

Next to the grid size, the gas compressibility has a large effect on the stiffness of the problem. Figs. 7 show the results for a more compressible (Fig. 7b) and a less compressible (Fig. 7b) case. The larger the degree of incompressibility, the higher the ratio between the imaginary part of the eigenvalues of the acoustic and convective modes. The effect of making the gas more incompressible is similar to refining the grid: the scaling of the imaginary part of the eigenvalues is linear in the speed of sound of the gas.

It is now interesting to see how the eigenvalues change when the first-order upwind spatial discretization, instead of the central discretization, is considered. Fig. 8 shows that the real part of the eigenvalues of both the acoustic and convective modes has grown enormously, indicating the numerical damping caused by the upwind scheme. The damping of the first order upwind method is in fact so large that almost all unstable modes have shifted to the left half of the complex plane. These observations are in line with Liao et al. (2008), who favour the central discretization for the accurate prediction of the stability of the two-fluid model.

Lastly, we consider the case where the superficial velocities are reduced so that the initial condition is in the well-posed stable regime. Fig. 9 shows that, employing once again the central scheme, all eigenvalues have a negative real part. This indicates a stable solution, in agreement with its position in the flow pattern map. The eigenvalues are organized in two ellipses, which is typical for travelling waves that are damped in time. Note that all the eigenvalues fall on the imaginary axis when the source terms are switched off (giving the inviscid two-fluid model) and the central scheme is used; this means that the solution consists of pure travelling waves. Thus, the friction terms in the equations are responsible for the damping of both the acoustic and the convective waves.

4.2. Stability properties of the time integration methods

For a stable time integration of the semi-discrete two-fluid model, the eigenvalues of the spatial discretization shown in Figs. 4–9 should fall inside the stability region of the time integration method. This stability region follows by application of the time integration method described by Eq. (31) to a linear test equation (as derived from Eq. (38)):

$$\dot{v}(t) = \mu v(t), \qquad \mu \in \mathbb{C}, \tag{39}$$

and computing the amplification factor:

$$G(z) = \left| \frac{v^{n+1}}{v^n} \right|, \quad z = \mu \Delta t. \tag{40}$$

This amplification factor indicates whether the numerical solution will grow in time. The exact solution to Eq. (39) is stable when $\mathrm{Re}(\mu) < 0$. A time integration scheme that inherits this property is called unconditionally stable or A-stable when stable solutions to the ODE (39) are obtained in a stable manner by the time integration method, for all choices of Δt. This can be expressed as:

$$\text{\$A\$-stable:} \quad G(z) \leq 1 \quad \text{for} \quad \mathrm{Re}(z) \leq 0. \tag{41}$$

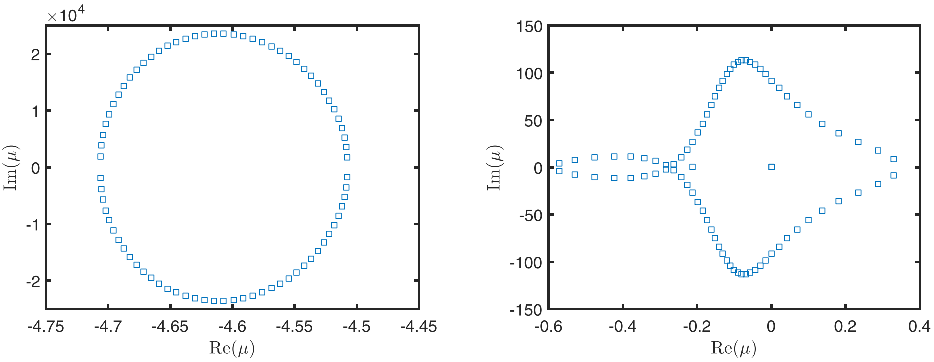
Next to A-stability, the concept of L-stability is important since we are dealing with a stiff system. L-stability indicates whether unresolved (very high) frequencies are damped by the numerical method. In terms of the amplification factor G it reads

$$\text{\$L\$-stable:} \quad G(z) \to 0 \quad \text{when} \quad \mathrm{Re}(z) \to -\infty. \tag{42}$$

Application of the time integration method expressed by Eq. (31) to the test Eq. (39) leads to the following equation for G:

$$(a_0 - \mu \Delta t \theta) G^2 + (a_1 - \mu \Delta t (1 - \theta)) G + a_2 = 0. \tag{43}$$

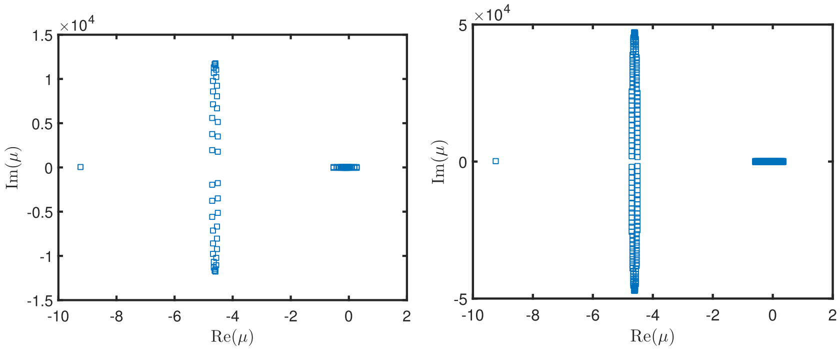
The amplification factor can be displayed as a contour in the complex $z = \mu \Delta t$ space (see any standard textbook on time integration methods, e.g. Butcher (2003); Hairer et al. (2008)). Here we take a slightly different approach: we set a certain time step and display G in complex μ space instead of $\mu \Delta t$ space. This allows direct comparison with the eigenvalues of the semi-discrete equations (see case I in Fig. 4). For the specific case that the time step is based on the liquid velocity ($\Delta t = 1/40$ s), and not on the speed of sound, Figs. 10a–10 c show the eigenvalue locations and the corresponding amplification factor. The vertical axis is on log-scale and only the positive imaginary part is shown (all eigenvalues appear as complex conjugates). Fig. 10 shows that all the three considered methods are A-stable, since $G < 1$ in the left half part of the complex plane. Crank–Nicolson is the only method which will give, independent of Δt, the correct unstable solutions to (39) for all z when $\mathrm{Re}(z) > 0$. For BDF2 and Backward Euler the time step has to be sufficiently small so that the eigenvalues move into the region $G > 1$. Since the two-fluid model with friction terms can exhibit unstable solutions (Fig. 5b), with eigenvalues in the right-half of the complex plane, this is a desirable feature.



(a) Acoustic.

(b) Convective.

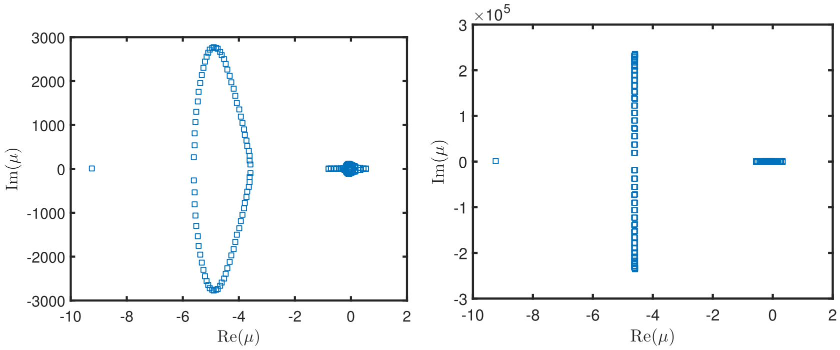
Fig. 5. Details of Fig. 4.



(a) IIA: $N = 20$.

(b) IIB: $N = 80$.

Fig. 6. Case II: grid size effects.



(a) IIIa: 0.1 $c_{g,0}$.

(b) IIIb: 10 $c_{g,0}$.

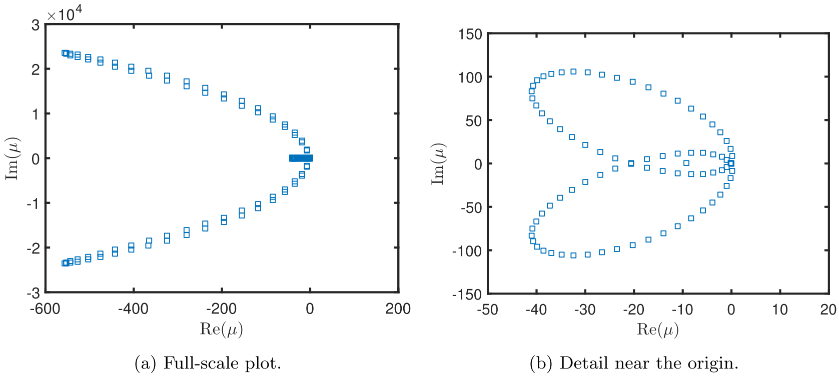
Fig. 7. Case III: compressibility effects.

However, the amount of damping that Crank–Nicolson has on the acoustic modes is not favourable. Consider the largest μ (in absolute sense), then the amplification factors for $\Delta t = 1/40$ s are:

$$G_{\text{exact}} = 0.8914, \qquad G_{\text{CN}} = 1.0000, \tag{44}$$

$$G_{\text{BDF2}} = 0.0304, \qquad G_{\text{BE}} = 0.00174. \tag{45}$$

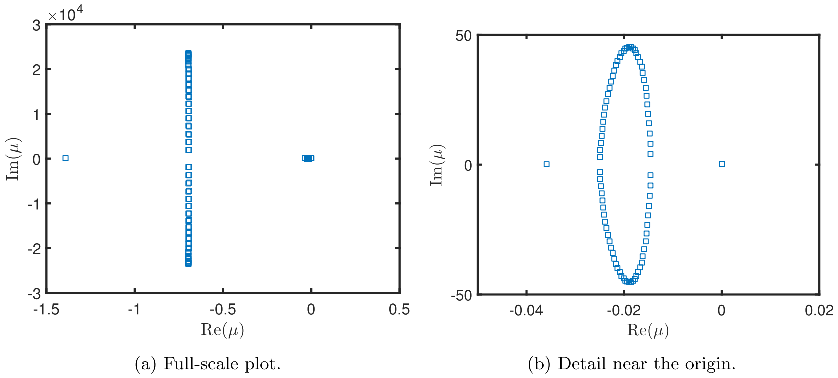
Although the amplification factor of Crank–Nicolson is closest to the exact solution (due to its lowest truncation error), it does not possess any damping. At a time step based on the liquid velocity, which in the 1 m long pipe can track the liquid velocity but not the acoustic velocity, the acoustic waves are not resolved and damping is a necessity to prevent unphysical oscillations.



(a) Full-scale plot.

(b) Detail near the origin.

Fig. 8. Eigenvalues for case IV: upwind discretization.



(a) Full-scale plot.

(b) Detail near the origin.

Fig. 9. Eigenvalues for case V: well-posed stable.

5. Von Neumann analysis on the fully discrete equations

5.1. Introduction

In the previous sections the eigenvalues of the continuous and semi-discrete equations were discussed. In the fully discrete case a stability analysis can be performed that is very similar to the one applied to the continuous case (Section 2.3). This is known as von Neumann stability analysis (Morton and Mayers, 2005), which can be seen as the discrete counterpart of the Kelvin–Helmholtz stability analysis (Akselsen, 2016). It is applicable in case of linear, constant coefficient partial differential equations with periodic boundary conditions. We employ therefore the linearized equations (37) in order to be able to apply the von Neumann analysis. As an example, we first consider the equations that result from applying the Backward Euler method:

$$\frac{\tilde{\boldsymbol{U}}^{n+1}-\tilde{\boldsymbol{U}}^{n}}{\Delta t}=\boldsymbol{J}\tilde{\boldsymbol{U}}^{n+1}, \tag{46}$$

where $\tilde{\boldsymbol{U}}$ contains the conservative variables defined on the staggered grid as shown in Fig. 3.

In the von Neumann analysis, which is essentially a discrete Fourier analysis (Morton and Mayers, 2005), the solution is assumed to consist of a finite sum of discrete waves, each with a different wave number:

$$\tilde{\boldsymbol{U}}_i^n=\sum_m(\boldsymbol{V}_i^n)_m, \tag{47}$$

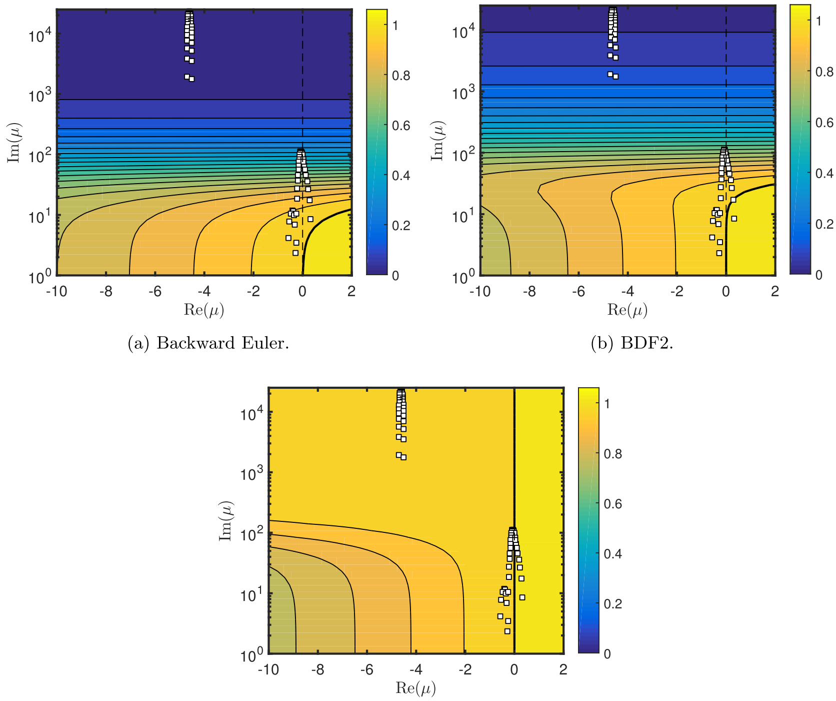
$$(\boldsymbol{V}_i^n)_m=e^{Ik_m s_i}\,\boldsymbol{v}_m^n=e^{Ik_m s_i}\,\boldsymbol{G}_m^n\,\hat{\boldsymbol{v}}_m, \tag{48}$$

where m indicates the mode number, k_m the wave number of mode m, $\boldsymbol{G}\in\mathbb{C}^{q\times q}$ the amplification matrix of mode m, and $\hat{\boldsymbol{v}}_m\in\mathbb{C}^q$ the Fourier coefficients of mode m of the initial condition. $q=4$ is the dimension of the problem. The fact that we employ an amplification matrix is similar to the approach followed by Fullmer et al. (2014), but slightly different from the scalar amplification factor considered by Liao et al. (2008). We will comment later on the relation between the two. Note that for simplified problems (such as a scalar convection-diffusion equation), the matrix $\boldsymbol{J}$ is circulant, which means that there is a direct connection between the discrete Fourier transform and the eigendecomposition (38). Unfortunately, the spatially discretized two-fluid model equations do not result in a matrix with circulant structure.

Since Eq. (46) is linear it suffices to consider a single mode $(\boldsymbol{V}_i^n)_m$ (consisting of solution components at grid points i and $i+1/2$ due to the staggering) to analyse the stability properties of the discretization method:

$$\begin{pmatrix}V_{1,i}\\V_{2,i}\\V_{3,i+1/2}\\V_{4,i+1/2}\end{pmatrix}_m^{n+1}-\begin{pmatrix}V_{1,i}\\V_{2,i}\\V_{3,i+1/2}\\V_{4,i+1/2}\end{pmatrix}_m^{n}=\Delta t\sum_{j=1}^{N}\boldsymbol{J}_{ij}\begin{pmatrix}V_{1,j}\\V_{2,j}\\V_{3,j+1/2}\\V_{4,j+1/2}\end{pmatrix}_m^{n+1}. \tag{49}$$

Since $\boldsymbol{J}$ contains the spatial discretization it is very sparse. Substituting the Fourier expansion (48) into Eq. (49), omitting the subscript m, and dividing each equation by the complex exponential



(a) Backward Euler.

(b) BDF2.

(c) Crank-Nicolson.

Fig. 10. Amplification factor G for different time integration methods when superimposed on the eigenvalues of Fig. 4. Contour lines are 0.05 apart; thick black lines indicate $G = 1$.

in space, yields the compact expression

$$\begin{pmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i+1/2} \\ v_{4,i+1/2} \end{pmatrix}^{n+1} - \begin{pmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i+1/2} \\ v_{4,i+1/2} \end{pmatrix}^{n} = \Delta t \boldsymbol{M}(e^{Ik_m s_i}) \begin{pmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i+1/2} \\ v_{4,i+1/2} \end{pmatrix}^{n+1}. \quad (50)$$

Eq. (50) is useful in a fully analytical point approach, whereas Eq. (49) is the equation that can be used when performing symbolic operations in our coding environment. The notation $\boldsymbol{M}(e^{Ik_m s_i})$ indicates the action of the spatial discretization on a complex exponential and is called the discrete Fourier symbol; it follows when substituting a discrete exponential in the discretization method and subsequently dividing it by this discrete exponential. For example, if a central discretization is used for the convective term in the mass equation for the gas, i.e.

$$V_{1,i}^{n+1} - V_{1,i}^{n} = -\Delta t \frac{V_{3,i+1/2}^{n+1} - V_{3,i-1/2}^{n+1}}{\Delta s}, \quad (51)$$

then the Fourier representation is

$$v_{1,i}^{n+1} - v_{1,i}^{n} = -\Delta t \frac{v_3^{n+1} e^{Ik_m s_{i+1/2}} - v_3^{n+1} e^{Ik_m s_{i-1/2}}}{e^{Ik_m s_i} \Delta s}, \quad (52)$$

and the matrix entry is

$$\boldsymbol{M}_{1,3} = -\frac{I \sin(k_m \Delta s/2)}{\Delta s/2}. \quad (53)$$

The full matrix is similar to the one presented by Liao et al. (2008), but much more elaborate due to the fact that we employ a 4 × 4 system including compressibility effects. In Section 5.3 a method will be proposed which circumvents the explicit formulation of this matrix. We write Eq. (50) in the following generic form, using a notation similar to Fullmer et al. (2014):

$$\boldsymbol{N} \boldsymbol{v}^{n+1} = \boldsymbol{O} \boldsymbol{v}^{n}, \qquad \text{or} \qquad \boldsymbol{v}^{n+1} = \boldsymbol{N}^{-1} \boldsymbol{O} \boldsymbol{v}^{n}, \quad (54)$$

where for Backward Euler: $\boldsymbol{N} = \boldsymbol{I} - \Delta t \boldsymbol{M}$, and $\boldsymbol{O} = \boldsymbol{I}$. The amplification matrix is given by

$$\boldsymbol{G} = \boldsymbol{N}^{-1} \boldsymbol{O}. \quad (55)$$

Stability depends on the spectral radius of $\boldsymbol{G}$, i.e. the maximum absolute value of the (complex) eigenvalues λ_G of $\boldsymbol{G}$. These eigenvalues λ_G follow from the eigenvalue problem

$$\det(\boldsymbol{N} \lambda_G - \boldsymbol{O}) = 0, \quad (56)$$

and a scalar amplification factor can be defined as $G = \max_i |\lambda_{G,i}|$. The amplification matrix $\boldsymbol{G}$ is the one used by Fullmer et al. (2014), whereas the amplification factor λ_G is used by Liao et al. (2008). $\boldsymbol{G}$ contains the growth and frequency components of all waves, which allows the reconstruction of the full solution, which is not possible with λ_G. In Section 5.3 we will obtain the amplification factors from simulation data only, and this requires the use of the full $\boldsymbol{G}$.

Once the amplification matrix $\boldsymbol{G}$ and the Fourier coefficients $\boldsymbol{v}$ of the initial condition are known, the solution at a time instance t^n in terms of Fourier coefficients is obtained from

$$\boldsymbol{v}_m^n = \boldsymbol{G}_m^n \hat{\boldsymbol{v}}_m, \quad \text{or} \quad \boldsymbol{v}_m^{n+1} = \boldsymbol{G}_m \boldsymbol{v}_m^n. \quad (57)$$

$\boldsymbol{G}$ contains information about the *growth* of the solution (diffusive errors), and about the *shift* (dispersive errors) of the solution in time.

5.2. Extension to BDF2 and Crank–Nicolson

For the BDF2 scheme, Eq. (50) becomes

$$\boldsymbol{N}\boldsymbol{v}^{n+1} + \boldsymbol{O}\boldsymbol{v}^{n} + \boldsymbol{P}\boldsymbol{v}^{n-1} = 0, \tag{58}$$

where $\boldsymbol{N} = a_0\boldsymbol{I} - \Delta t\boldsymbol{M}$, $\boldsymbol{O} = a_1\boldsymbol{I}$, $\boldsymbol{P} = a_2\boldsymbol{I}$. This equation can be written as

$$\begin{pmatrix} a_0\boldsymbol{I} - \Delta t\boldsymbol{M} & 0 \\ 0 & \boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{v}^{n+1} \\ \boldsymbol{v}^{n} \end{pmatrix} = \begin{pmatrix} -a_1\boldsymbol{I} & -a_2\boldsymbol{I} \\ \boldsymbol{I} & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{v}^{n} \\ \boldsymbol{v}^{n-1} \end{pmatrix},$$

$$\text{or:} \qquad \hat{\boldsymbol{N}}\boldsymbol{w}^{n+1} = \hat{\boldsymbol{O}}\boldsymbol{w}^{n}, \tag{59}$$

from which the definition of $\boldsymbol{G}$ follows:

$$\boldsymbol{G} = \hat{\boldsymbol{N}}^{-1}\hat{\boldsymbol{O}} = \begin{pmatrix} -a_1(a_0\boldsymbol{I} + \boldsymbol{M})^{-1} & -a_2(a_0\boldsymbol{I} + \boldsymbol{M})^{-1} \\ \boldsymbol{I} & 0 \end{pmatrix}. \tag{60}$$

This is a particular case of the more generic linear multistep methods analysed in Hundsdorfer and Verwer (2007). The eigenvalues of $\boldsymbol{G}$ follow from the determinant equation

$$\det(\hat{\boldsymbol{N}}\lambda_G - \hat{\boldsymbol{O}}) = \det((a_0\boldsymbol{I} - \Delta t\boldsymbol{M})\lambda_G^2 + a_1\lambda_G\boldsymbol{I} + a_2\boldsymbol{I}) = 0. \tag{61}$$

Note the similarity with the scalar amplification Eq. (43). When including the Crank–Nicolson method this can be generalized to

$$\det\left(\frac{a_0 + a_1G^{-1} + a_2G^{-2}}{\Delta t}\boldsymbol{I} - \left(\theta\boldsymbol{M} + (1-\theta)\boldsymbol{M}G^{-1}\right)\right) = 0, \tag{62}$$

where we write G instead of λ_G.

5.3. Amplification factor from simulation data

In this section we propose an alternative, novel method to obtain the von Neumann amplification factors without the need of doing symbolic manipulations, as this can be cumbersome for nonlinear models with elaborate closure relations, such as in the case of the two-fluid model. The idea is to perform a simulation with a small-amplitude harmonic as initial condition and to derive the amplification matrix by comparing the Fourier transform of the solution at a certain time level to the Fourier transform of solutions at previous time levels. We call this the *automatic* von Neumann analysis, in contrast to the classical *symbolic* von Neumann analysis that uses symbolic manipulations. First, the Fourier coefficients $\boldsymbol{v}_m$ of mode m are determined from the numerical solution $\boldsymbol{V}$ by a discrete Fourier transform, similar to Eq. (48):

$$\boldsymbol{v}_m^n = \sum_i \boldsymbol{V}_i^n e^{Ik_ms_i}\Delta s_i. \tag{63}$$

Given the solutions $\boldsymbol{v}^{n+1}$ and $\boldsymbol{v}^{n}$, the coefficients of the matrix cannot be determined from (57), since we have q equations for q^2 unknowns. We therefore perform q time steps and write

$$\begin{pmatrix} | & | & & | \\ \boldsymbol{v}_m^{n+1} & \boldsymbol{v}_m^{n} & \dots & \boldsymbol{v}_m^{n+2-q} \\ | & | & & | \end{pmatrix} = \tilde{\boldsymbol{G}}_m \begin{pmatrix} | & | & & | \\ \boldsymbol{v}_m^{n} & \boldsymbol{v}_m^{n-1} & \dots & \boldsymbol{v}_m^{n+1-q} \\ | & | & & | \end{pmatrix}, \tag{64}$$

or

$$\boldsymbol{Q}_m^{n+1} = \tilde{\boldsymbol{G}}_m\boldsymbol{Q}_m^{n}, \tag{65}$$

from which the matrix $\tilde{\boldsymbol{G}}_m$ can be readily determined:

$$\tilde{\boldsymbol{G}}_m = \boldsymbol{Q}_m^{n+1}(\boldsymbol{Q}_m^{n})^{-1}. \tag{66}$$

We write $\tilde{\boldsymbol{G}}$ instead of $\boldsymbol{G}$ to distinguish between the symbolic and the automatic von Neumann analysis. For the BDF2 scheme, $\tilde{\boldsymbol{G}} \in \mathbb{C}^{2q\times 2q}$, and the system of equations is extended to

$$\left(\begin{array}{cccc} | & | & & | \\ \boldsymbol{v}_m^{n+1} & \boldsymbol{v}_m^{n} & \dots & \boldsymbol{v}_m^{n+2-2q} \\ | & | & & | \\ \hline | & | & & | \\ \boldsymbol{v}_m^{n} & \boldsymbol{v}_m^{n-1} & \dots & \boldsymbol{v}_m^{n+1-2q} \\ | & | & & | \end{array}\right) = \tilde{\boldsymbol{G}}_m \left(\begin{array}{cccc} | & | & & | \\ \boldsymbol{v}_m^{n} & \boldsymbol{v}_m^{n-1} & \dots & \boldsymbol{v}_m^{n+1-2q} \\ | & | & & | \\ \hline | & | & & | \\ \boldsymbol{v}_m^{n-1} & \boldsymbol{v}_m^{n-2} & \dots & \boldsymbol{v}_m^{n-2q} \\ | & | & & | \end{array}\right). \tag{67}$$

In order to construct $\boldsymbol{Q}_m^{n+1}$ and $\boldsymbol{Q}_m^{n}$, it suffices to take $2q$ time steps and to store the Fourier coefficients at each time step (of course, one can also store the entire solution and calculate the Fourier coefficients afterwards). Once $\tilde{\boldsymbol{G}}$ is determined, the absolute value of its eigenvalues can be investigated. This has to be done for all wavenumbers, either by rerunning the simulation with an initial condition for each different wave, or performing one simulation with a single initial condition composed of all wavenumbers. This is a simple and fast procedure given that only $2q = 8$ time steps are necessary to reconstruct $\tilde{\boldsymbol{G}}$. In practice, we have noted that due to the high condition number of $\boldsymbol{Q}^n$ inaccurate results are sometimes obtained. This can be resolved by increasing the number of time instances in the analysis to for example $4q$, which works well in our simulations. The matrices $\boldsymbol{Q}^n$ and $\boldsymbol{Q}^{n+1}$ then become non-square and the solution of (67) should be interpreted in a least-squares sense.

To summarize, in this section we have explained two techniques, symbolic and automatic von Neumann analysis. This will be demonstrated for the Kelvin–Helmholtz instability case in Section 6.1. The first is the classic analysis: substitution of a Fourier series in the discretization matrix and investigating the resulting amplification matrix and its eigenvalues. We have done this by using the symbolic toolbox of Matlab and by direct substitution of the complex exponentials into our code to arrive at symbolic expressions for the amplification matrix. The second technique is based on substituting sinusoidal wave perturbations in the initial conditions, and running the code for several time steps. This can be used to check the outcome of the first technique, but also to obtain amplification matrices for black-box solvers (for instance commercial codes).

6. Results for various test cases

In this section we report the results of four test cases. The first three are related to the Kelvin–Helmholtz instability: the first one considers steady state flow with a small but unstable perturbation in a single wave to validate the linear stability (von Neumann) and accuracy of the time integration methods. The second case considers the same instability but with a much larger perturbation to study the nonlinear wave growth and the identification of ill-posedness. The third case considers stiffness by including perturbations in both the convective and acoustic waves, and we investigate how the time integrators resolve the acoustics. Lastly, in the fourth test case the propagation of a hold-up wave is investigated, and all the previously investigated concepts of stability, accuracy, ill-posedness and resolution of acoustics are considered. The four cases will be referred to as A, B, C, and D, respectively.

6.1. Kelvin–Helmholtz: Linear wave growth

6.1.1. Modified wave number analysis

We consider the Kelvin–Helmholtz instability for the test problem described in Section 2.4 and with the conditions given in Table 1. The exact solution to the linearized system of equations (19) is given by

$$\boldsymbol{W}(s,t) = \boldsymbol{W}_0 + \sum_j \mathrm{Re}\left[\boldsymbol{\varepsilon}_j e^{I(\omega_j t - ks)}\right]. \tag{68}$$

The initial condition is obtained by a small perturbation in the liquid hold-up: $\tilde{\alpha}_l = 10^{-6}$. The perturbation vector $\boldsymbol{\varepsilon}_3$ is taken to be the eigenvector associated to the angular frequency ω_3 (Liao et al., 2008):

$$\boldsymbol{\varepsilon}_3 = 10^{-4}\begin{pmatrix} 1\cdot 10^{-2} \\ 7.005\cdot 10^{-3} - 1.1025\cdot 10^{-3} I \\ 2.497\cdot 10^{-1} + 1.186\cdot 10^{-3} I \\ -3.619 - 6.550\cdot 10^{-1} I \end{pmatrix}. \tag{69}$$

First we perform a consistency check: simultaneous grid and time step refinement for a fixed wave number $k = 2\pi$ with

$$\mathcal{C}_l \approx 1 \rightarrow \Delta t \approx \Delta s = \frac{1}{N}, \tag{70}$$

where $\mathcal{C}$ is the dimensionless time step, similar to the CFL number (which strictly is a stability condition). $k = 2\pi$ is the smallest wave number that can presented on our periodic simulation domain. The exact growth rate, $\mathrm{Im}(\omega_3) = -0.35$, is compared to the numerically computed growth rates as obtained from the symbolic and the automatic von Neumann analysis:

$$\omega_{\mathrm{vN}} = \frac{\ln\left(\min\left(\mathrm{Im}(\lambda(\boldsymbol{G}_m))\right)\right)}{\Delta t}, \qquad \tilde{\omega}_{\mathrm{vN}} = \frac{\ln\left(\min\left(\mathrm{Im}\left(\lambda(\tilde{\boldsymbol{G}}_m)\right)\right)\right)}{\Delta t}. \tag{71}$$

Even though only a single wave is triggered due to the initial perturbation in $\boldsymbol{\varepsilon}_3$, the discrete amplification matrix $\boldsymbol{G}$ still has four eigenvalues. This is because the Fourier transform of the initial condition does not consist of a single wave, but of four waves. We are interested in the one that is largest in magnitude, since it indicates whether the numerical solution is stable or not. However, in contrast to the classical von Neumann analysis, in this study unstable solutions are not necessarily unwanted, since the differential equation itself has an unstable behaviour (Fullmer et al., 2014), which might be associated to slug flow.

Fig. 11 shows the comparison of ω_3 to ω_{vN} and $\tilde{\omega}_{\mathrm{vN}}$, for different grids and discretization schemes. Fig. 11a shows that all time discretization methods converge to the exact growth rate upon grid refinement. However, for coarse grids, the Backward Euler method predicts a positive ω (damped solutions), whereas BDF2 and Crank–Nicolson correctly predict growing solutions (albeit with a reduced growth rate). In Fig. 11b this is made more quantitative by computing the error of the discrete models compared to the exact value:

$$\eta = |\omega_3 - \omega_{\mathrm{vN}}|. \tag{72}$$

It is clear that the use of Backward Euler or the use of the first order upwind scheme limits the accuracy to first order. Fig. 11b also indicates that for very fine meshes there is a slight discrepancy between the symbolic and automatic von Neumann analysis. This is related to the matrix inversion required for the reconstruction (66) in the automatic von Neumann analysis, which can suffer from numerical inaccuracies. For the main purpose of the von Neumann analysis, namely investigating the behaviour of the numerical growth rate or dispersion error as function of phase angle ($\phi = k\Delta s$), this is not an issue.

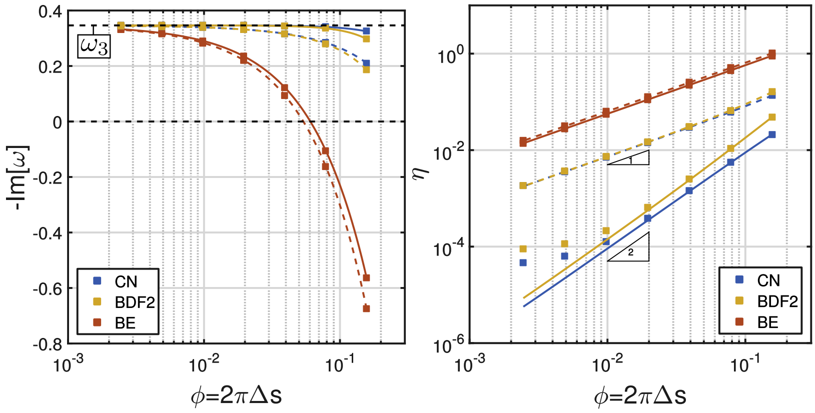
Secondly, we investigate how waves grow in time, depending on the wave number k and the spatial and temporal discretization. The number of grid points is fixed ($\Delta s = 1/160$ m). The shortest wavelength that can be represented on the grid is $2\Delta s$ (Nyquist limit), corresponding to the wavenumber $k = \pi/\Delta s$, and phase angle $\phi = \pi$. The wave number analysis provides insight into how well waves of different frequency are resolved by the time integration method. It is similar to the analysis for spatial discretization methods done by Liao et al. (2008). Fig. 12a shows the growth rate $G = \max|\lambda(\boldsymbol{G})|$ which compares well with the results of Liao et al. (2008). In addition, Fig. 12b shows the growth rate ω_{vN} instead of the amplification factor, which includes the exact solution (denoted by the black dashed line). This line corresponds to the dashed line previously shown in Fig. 2b. It is perhaps not surprising to see that Crank–Nicolson ('central in time') with a central scheme in space leads to an accurate prediction of G. In fact, when performing the analysis without friction terms (in the well-posed stable regime), the Crank–Nicolson / central combination leads to $G = 1$ *independent of the wavenumber*. This is a well-known result for advection equations and also holds for the two-fluid model without source terms. However, the presence of source terms leads to unstable solutions; their growth rate can be captured by the numerical scheme though, with the accuracy of the time integration method. Furthermore, $G = 1$ does not mean that the numerical is exact, but that there are only dispersive errors and no diffusive errors.

For each wavenumber k the corresponding growth rate ω_3 is negative. This agrees with the stability-hyperbolicity theorem (Prosperetti and Tryggvason, 2007) which says that the neutral stability limit is independent of the wavenumber. The case of $k = 2\pi$ from Fig. 11 is highlighted as the black vertical line in Fig. 12. For $\phi < 10^{-2}$ all numerical methods perform well - this corresponds to long, low frequency waves which are well resolved. For larger ϕ the numerical damping of all methods becomes apparent. The kinks in the amplification factor and the growth rate are due to the fact that a different eigenvalue (see Eq. (71)) becomes dominant.

6.1.2. Linearized discrete flow pattern map prediction

In this section we propose a novel way for displaying the performance of numerical methods, which we call 'Discrete Flow Pattern Maps' (DFPM), that uses the growth rate determination method developed for the automatic von Neumann analysis (Eq. (66)). Whereas traditional flow pattern maps, such as the one in Fig. 1, display stable and unstable regimes based on the properties of the differential equations, the DFPM displays the *effective* stability regions that result when the discrete equations are solved, with a certain numerical method and a certain grid and time resolution. Such a map is of crucial importance as an indication whether a discretization method is able to correctly capture the well-posed unstable regime (and the potential transition to slug flow) or whether numerical diffusion overwhelms the physical growth of instabilities.

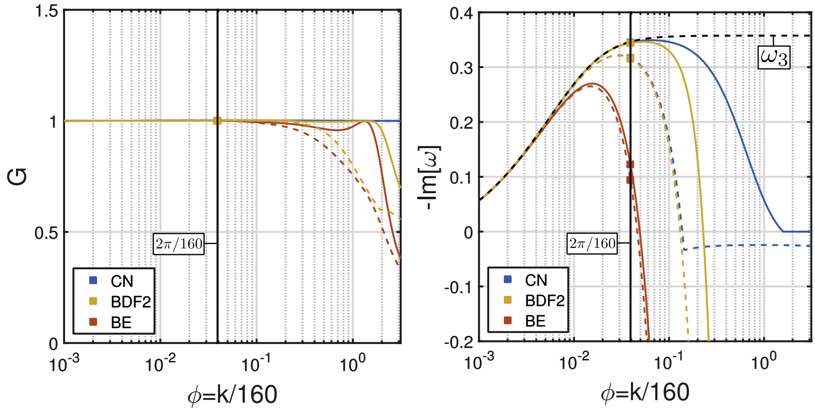
The DFPM can be constructed in the same way in which we constructed the flow pattern map in Fig. 1. We employ a sequence of superficial liquid and gas velocities, solve the discrete equations with a small perturbation as initial condition and determine the growth rate $\tilde{\omega}_{\mathrm{vN}}$ from (66) and (71). The stability boundary is given by $\mathrm{Im}(\tilde{\omega}_{\mathrm{vN}}) = 0$. Fig. 13a shows the VKH stability boundary for Backward Euler and BDF2 for two different grids (and associated time steps). It can be seen that BDF2 captures the exact stability boundary very accurately on all grids. On the other hand, the effective stability region given by Backward Euler is much larger (note the log scale) than the actual stability region of the differential equations, due to the large amount of artificial diffusion added by Backward Euler. For example, for $N = 40$, and at a superficial



(a) Growth rate.

(b) Error in growth rate.

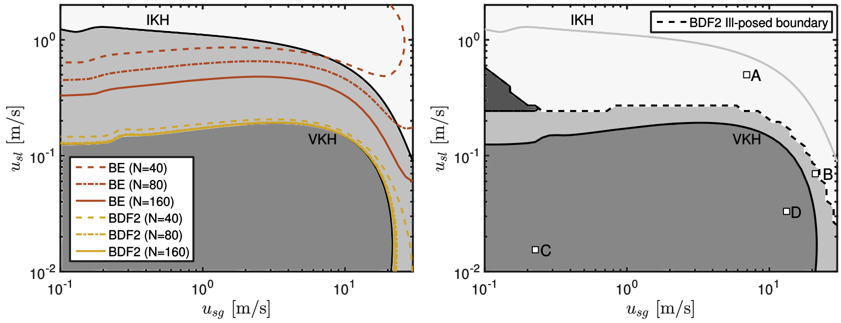
Fig. 11. Comparison of the growth rate and the error in the growth rate for different discretization methods as a function of the grid size. Squares: automatic von Neumann analysis, lines: symbolic von Neumann analysis. Dashed lines: first order upwind, solid lines: second order central.



(a) Amplification factor.

(b) Growth rate.

Fig. 12. Comparison of the growth rate and the error in the growth rate for different discretization methods as a function of the wave number. Dashed lines: first order upwind, solid lines: second order central.



(a) Effect of time integration methods and resolution.

(b) Effect of nonlinear growth.

Fig. 13. Discrete flow pattern maps (DFPM).

gas velocity of $u_{sg} = 10$ m/s, the two-fluid model predicts growing waves at a superficial liquid velocity of $u_{sl} = 0.15$ m/s, but with Backward Euler the wave growth is only apparent when the superficial liquid is increased to $u_{sl} = 0.6$ m/s (already at the ill-posed boundary).

Any simulation starting above the ill-posedness boundary will directly be marked as ill-posed, because our ill-posedness indicator is based on the evaluation of the eigenvalues of the differential equations, see Eq. (12). However, it is possible that, due to nonlinear effects, simulations that start in the well-posed unstable regime grow into the ill-posed regime. This is investigated in the next section.

6.2. Kelvin–Helmholtz: Nonlinear wave growth

In this section the simulation from Section 6.1 is repeated, but with a larger perturbation: $\tilde{\alpha}_l = 10^{-2}$, and $\tilde{\boldsymbol{\varepsilon}}_3 = 10^4 \boldsymbol{\varepsilon}_3$. This causes the simulation to quickly enter the nonlinear regime, in which waves steepen, possibly leading to slug formation but potentially also to ill-posedness. Based on the results of the previous section, we select BDF2 with $N = 80$ to assess the growth of initial perturbations into the nonlinear regime and whether this leads to ill-posed results. Note that this is different from a previous paper (Hendrix et al., 2016), in which we have assessed the 'time to ill-posedness' as a metric to compare different time integration methods. In the present study the focus is on whether well-posed unstable solutions can be obtained for long time integration periods.

Similar to the linear case, we perform simulations with the two-fluid model with a central discretization for a range of superficial gas and liquid velocities but now until $t = 100$ s and only with a central discretization of the convective terms. Ill-posedness is investigated by checking if the eigenvalues of the differential equation are real or complex at the conditions predicted by the numerical simulation. If a complex eigenvalue occurs at any point in space or time the corresponding point in the flow pattern map is marked ill-posed. Fig. 13b shows that the resulting numerical ill-posedness boundary has shifted significantly into the well-posed unstable regime as compared to the ill-posedness boundary of the differential equations (indicated by IKH in Fig. 13b). It appears that *a large part of the well-posed unstable regime of the flow pattern map gives ill-posed solutions when actual numerical simulations are performed*. An example of a simulation which turns ill-posed is case A in Fig. 13b, which corresponds to the conditions studied in Section 6.1 for linear perturbations. The nonlinear behaviour of the hold-up fraction α_l in space and time is shown in Fig. 14 for this case. The solution becomes ill-posed already after approximately 5 s. The liquid hold-up fraction as a function of time at $s = 1$ m is shown in Fig. 15a.

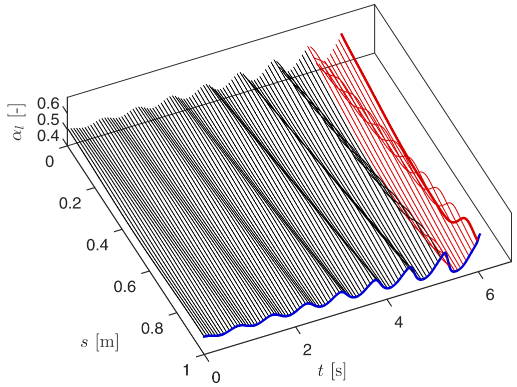


Fig. 14. Solution in space and time for case A, simulated with BDF2: well-posed unstable solution becoming ill-posed (indicated in red). The blue curve is shown in Fig. 15a. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

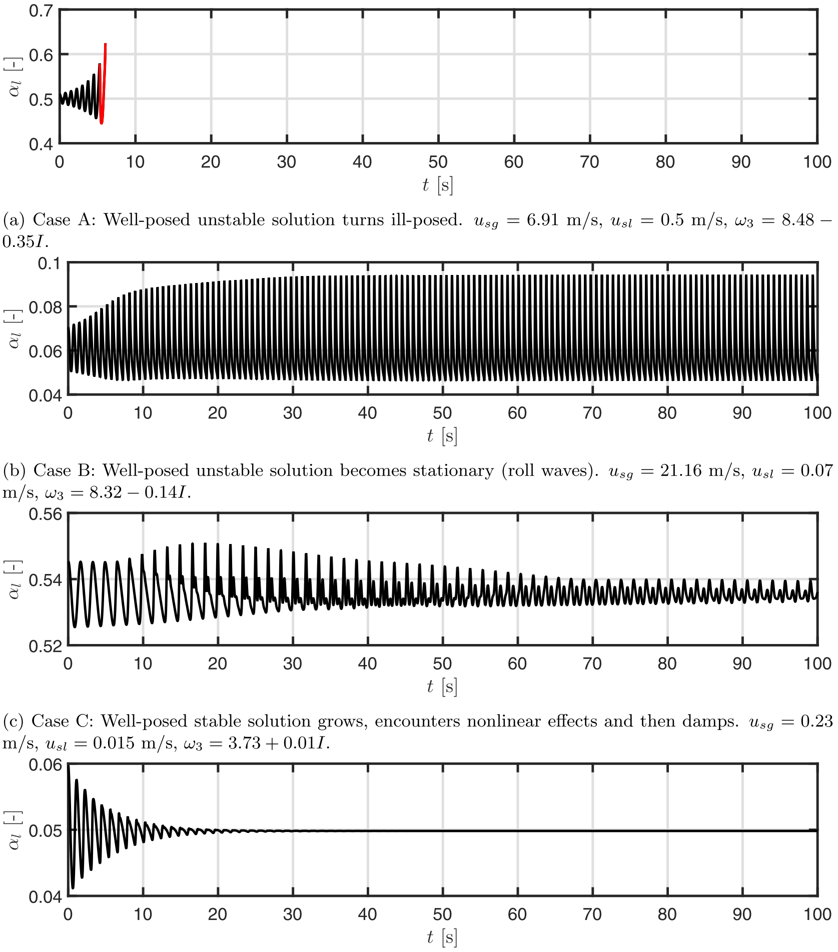
In addition to case A, three other cases (B, C, and D) are indicated in Fig. 13b which exhibit qualitatively different solution behaviour. Case B in Fig. 15b starts in the well-posed unstable regime, like case A, but after initial growth (as predicted by linear theory) stabilizes and forms a wave with a constant amplitude and frequency. Its shape and position in the flow pattern map indicate that this could be a so-called continuous 'roll wave': a particular solution to the two-fluid model which is constant in a reference frame moving with the flow (Barnea and Taitel, 1993; Holmås, 2010; Johnson, 2005). To check that this is not a numerical artefact, we have confirmed the roll wave presence with a simulation on a much finer grid ($N = 640$). Case C and D are both in the well-posed stable regime, where initial perturbations are damped according to the linear theory. This happens indeed for case D. However, for case C, the damping is very small ($\mathrm{Im}(\omega_3) = 0.01$) and nonlinear effects lead to wave growth and the appearance of new harmonics. The oscillation frequency is now lower because the real part of ω_3 has decreased. Within the time period of 100 s displayed in the plot it is unclear if the wave damps out. Continued simulation until 1000 s reveals that the wave eventually damps out, like in case D.

We note that the current simulation results are obtained based on a number of simplifications compared to pipeline simulations on actual geometries. First, we employed initial perturbations according to the eigenvector $\boldsymbol{\varepsilon}_3$. This means that the perturbations immediately grow according to linear theory until nonlinear effects take over. Second, we used a single wave with wavenumber $k = 2\pi$. Other wavenumbers will have different frequencies and growth rates and can lead to qualitatively different behaviour (although the VKH and IKH boundaries are independent of wavenumber). Furthermore, the periodic boundary conditions and size of the domain limit the frequencies that can be represented on the domain. Lastly, we note the black region in Fig. 13b, which indicates the region where the liquid reaches the top of the pipeline. This is not necessarily ill-posed, but outside the scope of our investigation.

6.3. Kelvin–Helmholtz: Acoustic wave problem

In this test the Kelvin–Helmholtz instability problem from Section 6.2 is extended to take into account acoustic waves, which were not present in previous studies (e.g. Fullmer et al., 2014; Liao et al., 2008) as these used the incompressibility assumption. We propose this acoustic wave problem as test case for pipeline simulation codes that solve the compressible two-fluid model. In order to demonstrate the acoustic filtering capabilities of BDF2 we choose two waves (see Eq. (23)): (i) a slow, right moving, unstable (growing) wave with frequency $\omega_3 = 8.48 - 0.35I$, and (ii) a fast, left moving, stable (damped) wave with frequency $\omega_1 = -1758.05 + 4.51I$. This extends Section 6.1 where only a single wave was considered. The hold-up fraction for the slow wave is again 10^{-6}, whereas the hold-up fraction for the fast wave is 10^{-2}. As before, the perturbations $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\varepsilon}_3$ are taken as the eigenvectors that follow from the dispersion analysis:

$$\boldsymbol{\varepsilon}_1 = \begin{pmatrix} 1 \cdot 10^{-6} \\ -5.616 \cdot 10^{-4} + 1.435 \cdot 10^{-6} I \\ -4.624 \cdot 10^{-1} + 4.876 \cdot 10^{-3} I \\ 1.577 \cdot 10^{2} - 1.276 I \end{pmatrix},$$



(a) Case A: Well-posed unstable solution turns ill-posed. $u_{sg} = 6.91$ m/s, $u_{sl} = 0.5$ m/s, $\omega_3 = 8.48 - 0.35I$.

(b) Case B: Well-posed unstable solution becomes stationary (roll waves). $u_{sg} = 21.16$ m/s, $u_{sl} = 0.07$ m/s, $\omega_3 = 8.32 - 0.14I$.

(c) Case C: Well-posed stable solution grows, encounters nonlinear effects and then damps. $u_{sg} = 0.23$ m/s, $u_{sl} = 0.015$ m/s, $\omega_3 = 3.73 + 0.01I$.

(d) Case D: Well-posed stable solution damps in time. $u_{sg} = 13.28$ m/s, $u_{sl} = 0.033$ m/s, $\omega_3 = 5.35 + 0.18I$.

Fig. 15. Numerical solutions for cases A–D as indicated in Fig. 13b, simulated using BDF2.

$$\boldsymbol{\varepsilon}_3 = \begin{pmatrix} 1 \cdot 10^{-2} \\ 7.005 \cdot 10^{-3} - 1.1025 \cdot 10^{-3} I \\ 2.497 \cdot 10^{-1} + 1.186 \cdot 10^{-3} I \\ -3.619 - 6.550 \cdot 10^{-1} I \end{pmatrix}. \tag{73}$$

The particular choice for hold-up fractions leads to the following combined wave system:

- A slow, right moving, growing liquid hold-up fraction wave.
- A fast, left moving, pressure wave, which is damped in time, so that after long periods a slow, right moving, growing pressure wave becomes dominant.
- A slow, right moving, growing wave in the liquid velocity, perturbed by a smaller amplitude, fast left moving wave, which is damped out.
- A fast, left moving, damped wave in the gas velocity, perturbed by a smaller amplitude, slow right-moving, growing wave, which becomes dominant for long periods.

A qualitative impression of the behaviour of the numerical solution for short and long time periods, for different integration methods, is given in Figs. 16–18. A central scheme has been used to discretize the convective terms. The short time scale is closely related to the speed of sound in the gas and we define a dimensionless time step of $\frac{1}{2}$ based on this time scale:

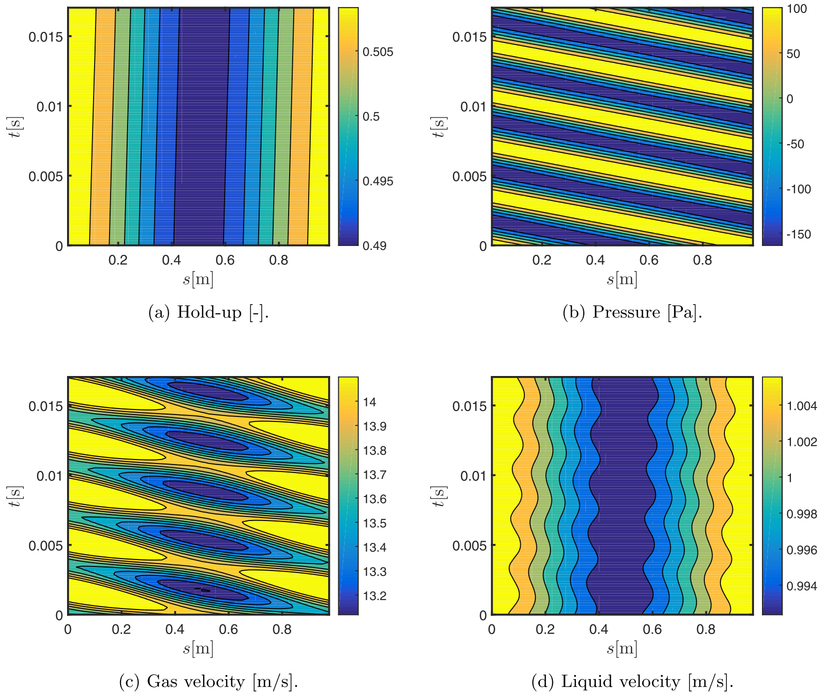
$$\mathcal{C}_c = \frac{1}{2} \rightarrow \Delta t = \frac{1}{2}\frac{\Delta s}{c_g} \approx 4.26 \cdot 10^{-5}\,\text{s}, \qquad t_{\text{end},c} = 400\Delta t \approx 0.017\,\text{s}. \tag{74}$$

The long time period is based on the initial liquid velocity, and we define a dimensionless time step and end time based on this time scale:

$$\mathcal{C}_l = 1 \rightarrow \Delta t = \frac{\Delta s}{u_l} = 0.025\,\text{s}, \qquad t_{\text{end},l} = 0.25\,\text{s}. \tag{75}$$

Note that the gas velocity is around 14 m/s, which gives a dimensionless time step based on the gas velocity of approximately 14.

First, we investigate the resolution of the acoustic waves on the short time scale. Fig. 16 displays the solution obtained with BDF2 until $t_{\text{end},\,c}$. The liquid hold-up fraction α_l clearly shows the slow



(a) Hold-up [-].

(b) Pressure [Pa].

(c) Gas velocity [m/s].

(d) Liquid velocity [m/s].

Fig. 16. Solution of acoustic wave problem in $s-t$ diagrams. BDF2, $\mathcal{C}_c$=0.5.

right moving wave associated with ω_3. The pressure, on the other hand, displays the fast wave moving to the left. The liquid and gas velocity show predominantly the left and right moving wave, respectively, but with perturbations in the other wave. The solution is quantitatively the same as the solutions obtained with the Crank–Nicolson scheme and with the Runge–Kutta method (reference solution) at this $\mathcal{C}_c = \frac{1}{2}$: both BDF2 and Crank–Nicolson are well able to resolve acoustic waves at a time step based on the speed of sound in the gas.

Secondly, we increase the time step and take $\mathcal{C}_l = 1$, which is common in many practical computations in which the accurate resolution of acoustic waves is not important. Fig. 17 shows that BDF2 effectively kills the acoustic waves during the first time steps, and only the hold-up wave remains. This corresponds to the amplification factor (45) and the results in Fig. 10b. On the other hand, Fig. 18 shows that when Crank–Nicolson is employed with this increased time step, a non-physical oscillation in the pressure appears which is not damped in time. This oscillation also becomes apparent in the gas velocity and to a lesser extent in the liquid velocity. The hold-up wave is hardly affected. Therefore, although Crank–Nicolson gives stable results (it is an A-stable method), it does not damp out unresolved transients (it is not L-stable). This confirms the results from Eq. (44) and in Fig. 10c.

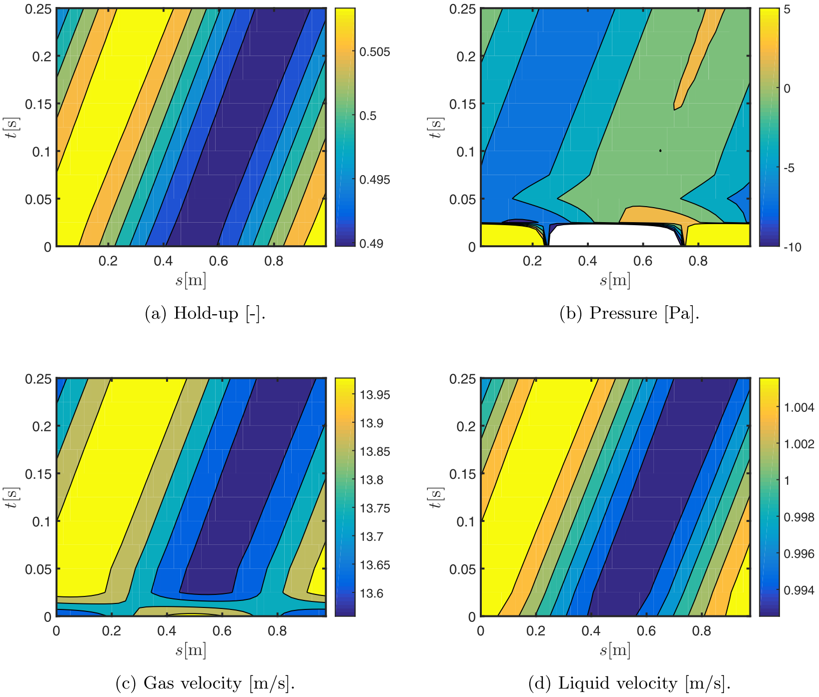
Fig. 19 compares the results for different time integration methods in a single plot at $t = 1$ s and $\mathcal{C}_l = 1$. The reference solution has been obtained with the explicit Runge–Kutta method with $\mathcal{C}_c = 0.5$ and its temporal error can be considered negligible. The BDF2 and CN schemes are both giving an accurate prediction for the liquid hold-up fraction and the liquid velocity. However, in both the pressure and the gas velocity, the prediction of Crank–Nicolson is very far off. This renders the Crank–Nicolson scheme essentially useless in these situations. Slightly increasing the value of θ (for example to 0.55), will improve the long time behaviour since some damping is introduced. However, this does not work on short time scales, and comes at a price of reduced accuracy. Backward Euler does not suffer from oscillations, though it has severely damped the amplitude of the wave. Overall, BDF2 performs best for this test case since it has the best balance between acoustic damping properties and accuracy.

7. Conclusions

In this paper we have analysed several time integration methods for the compressible two-fluid model with the goal to simulate stratified wavy flow and slug flow in pipelines in a so-called ‘slug capturing’ approach. The study was focussed on obtaining insight into the numerical pitfalls and requirements. Thereto a theoretical analysis on the differential equations and the discretized equations was given, in particular with respect to their stability properties. The analysis has been demonstrated for a number of test cases.

We conclude that the BDF2 method is a robust time integrator for the two-fluid model and it outperforms the commonly used Backward Euler method and the second order Crank–Nicolson method. This is due to a combination of its second order accuracy, A-stability and L-stability. These properties make that BDF2 can be used for the time integration of unsteady problems with a CFL number of 1 based on the liquid velocity, while not suffering from numerical oscillations that arise from acoustic wave propagation.

To facilitate the comparison and understanding of the time integration methods, several techniques have been proposed which have not been applied to the two-fluid model before. First, the eigenvalues of the spatial discretization have been investigated and it is shown how the stiffness depends linearly on the speed of



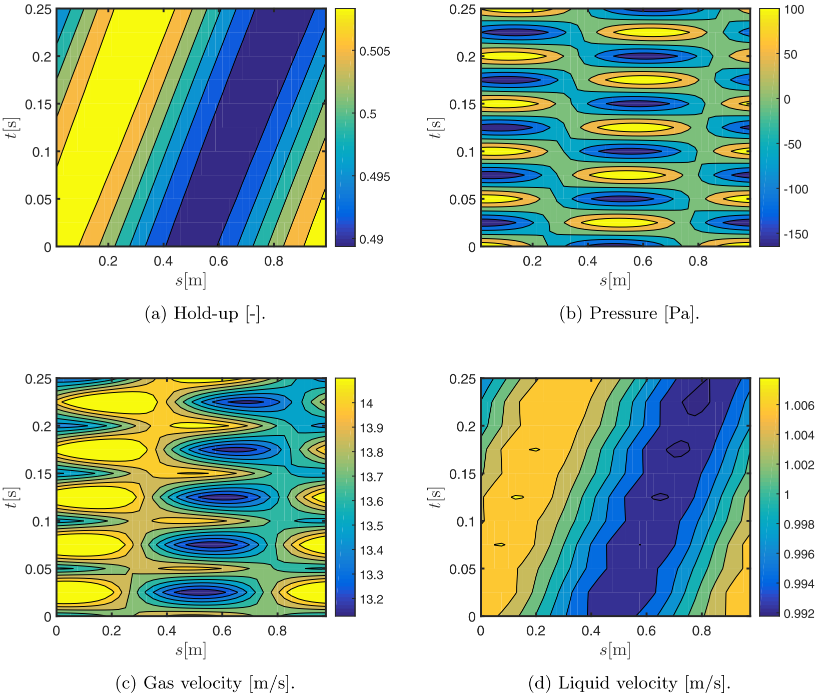
(a) Hold-up [-].

(b) Pressure [Pa].

(c) Gas velocity [m/s].

(d) Liquid velocity [m/s].

Fig. 17. Solution of acoustic wave problem in $s-t$ diagrams. BDF2, $\mathcal{C}_l$=1.



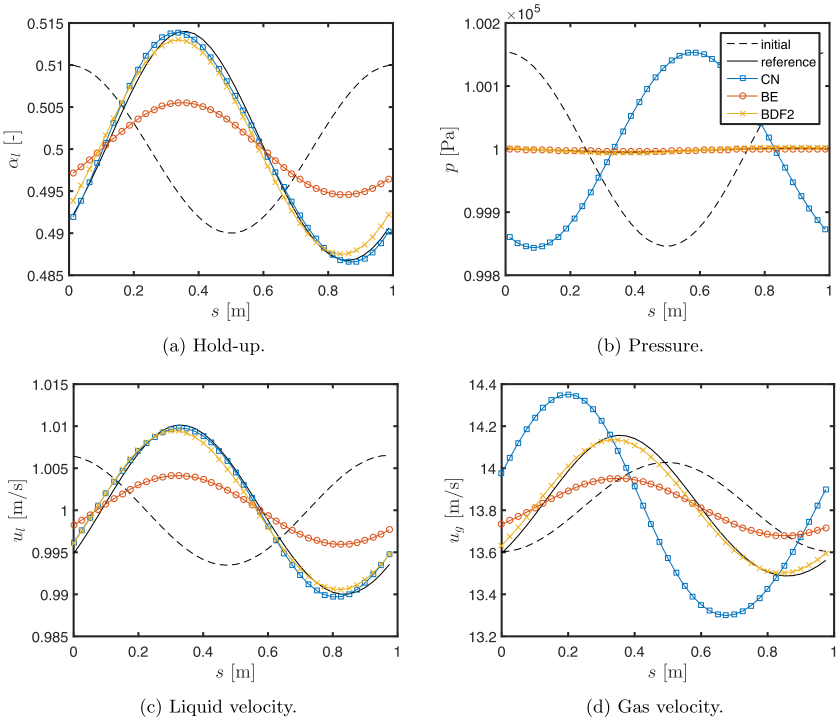
(a) Hold-up [-].

(b) Pressure [Pa].

(c) Gas velocity [m/s].

(d) Liquid velocity [m/s].

Fig. 18. Solution of acoustic wave problem in $s-t$ diagrams. Crank–Nicolson, $\mathcal{C}_l$=1.



(a) Hold-up.

(b) Pressure.

(c) Liquid velocity.

(d) Gas velocity.

Fig. 19. Comparison of time integration methods at $t = 1$ s for double wave problem with $\mathcal{C}_l = 1$.

sound and on the number of grid points. The central scheme is more accurate than the upwind scheme and captures the unstable region accurately. The comparison of the stability regions of the time integration methods shows the difference in damping properties between Backward Euler or BDF2 and Crank–Nicolson. Second, a new automatic von Neumann analysis technique has been developed as tool for direct evaluation of the stability of the discrete models by running a computer code without requiring symbolic manipulations. This makes it very flexible since it can be directly applied when more physics (e.g. surface tension or axial diffusion) or other spatial discretization methods are included. Third, we have proposed the use of Discrete Flow Pattern Maps (DFPM) to indicate to what extent discretization methods (for a certain choice of the number of grid points and the size of the time step) are able to reproduce the flow pattern maps that are based on the stability of the differential equations.

Several test cases have been used in which these techniques are used to compare BDF2 to other time integration methods: linear, nonlinear and acoustic Kelvin–Helmholtz instabilities. Accuracy tests and von Neumann analysis have shown its second order accuracy and comparable stability properties as Crank–Nicolson. BDF2 is slightly more diffusive than Crank–Nicolson but much more accurate than Backward Euler. The acoustic wave test shows that BDF2 is able to give accurate solutions, whether acoustic waves are resolved or not, whereas Crank–Nicolson leads to non-physical solutions. The ability of BDF2 to handle unresolved transients will also be important when dealing with other types of stiffness, for example caused by terms with small time scales (e.g. phase transition) or algebraic equations (e.g. the drift-flux model).

The Discrete Flow Pattern Map reveals that the *effective* well-posed unstable region is well captured by BDF2 but completely missed by Backward Euler, at least for the considered grid and time step resolutions. Simulations in the nonlinear regime furthermore have indicated that ill-posedness can occur when starting from the unstable regime. The implication is that the actual well-posed unstable regime for nonlinear simulations can be much smaller than the theoretical one, which can limit the application of the two-fluid model for simulating the stratified-slug flow transition.

We note that the DFPM as presented here is based on simulations in an idealized setting, with a simple geometry, initial conditions, and boundary conditions. In future work we plan to employ the BDF2 method to further study the formation of slug flow and ill-posedness issues under actual pipeline conditions.

In this paper we have relied on relatively standard spatial discretization techniques, since the focus is on the time discretization methods. Improvements in the spatial discretization (for example a method based on a Roe linearization) are needed to resolve discontinuities accurately and to capture the transition to locally single phase (slug) flow. Since the analysis in this paper does not require special properties of the spatial scheme, we expect that BDF2 can be applied to such discretization schemes without difficulty.

Acknowledgements

The authors would like to thank Ruud Henkes for the fruitful discussions and suggestions. Furthermore, the work by the second and third author was made possible by a grant from Shell Projects & Technology, which is greatly appreciated.

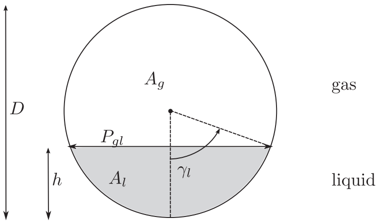


Fig. A.20. Stratified flow layout and definitions.

Appendix A. Two-fluid model details

A.1. Geometry

The following geometric identities are used to express the wall perimeters, interfacial perimeter, and liquid height in terms of the wetted angle γ_l (Fig. A.20):

$$P_{gl} = D \sin \gamma_l, \tag{A.1}$$

$$P_l = D\gamma_l, \tag{A.2}$$

$$P_g = D(\pi - \gamma_l), \tag{A.3}$$

$$h = \frac{1}{2}D(1 - \cos \gamma_l). \tag{A.4}$$

We use Biberg's approximation (Biberg, 1999) to express α_l in terms of γ_l (this avoids the iterative solution of a nonlinear equation):

$$\gamma_l = \pi \alpha_l + \left(\frac{3\pi}{2}\right)^{\frac{1}{3}} \left(1 - 2\alpha_l + \alpha_l^{\frac{1}{3}} - \alpha_g^{\frac{1}{3}}\right). \tag{A.5}$$

A.2. Friction models

The wall (subscript w) and interfacial (subscript gl) shear stress are expressed by the Fanning friction factor definition:

$$\tau = \begin{cases} \frac{1}{2} f_\beta \rho_\beta u_\beta |u_\beta| & \text{wall} \\ \frac{1}{2} f_{gl} \rho_g (u_\beta - u_\gamma)|u_\beta - u_\gamma| & \text{interfacial} \end{cases} \tag{A.6}$$

The friction factor f_β of phase β with the pipe wall is modeled with the Churchill relation (Churchill, 1977):

$$f_\beta = 2\left(\left(\frac{8}{\mathrm{Re}_\beta}\right)^{12} + (A + B)^{-1.5}\right)^{\frac{1}{12}}, \tag{A.7}$$

$$A = \left(2.457 \ln\left(\left(\left(\frac{7}{\mathrm{Re}_\beta}\right)^{0.9} + 0.27\frac{\epsilon}{D_{h\beta}}\right)^{-1}\right)\right)^{16}, \tag{A.8}$$

$$B = \left(\frac{37530}{\mathrm{Re}_\beta}\right)^{16}. \tag{A.9}$$

Here ϵ is the hydraulic pipe roughness, Re_β is the Reynolds number,

$$\mathrm{Re}_\beta = \frac{\rho_\beta u_\beta D_{h\beta}}{\mu_\beta}, \tag{A.10}$$

and $D_{h\beta}$ is the hydraulic diameter:

$$D_{h\beta} = \begin{cases} \frac{4A_l}{P_{lw}} & \text{if } \beta = l \\ \frac{4A_g}{P_{gw} + P_{gl}} & \text{if } \beta = g \end{cases} \tag{A.11}$$

The interfacial friction factor f_{gl} is calculated by (Liao et al., 2008):

$$f_{gl} = \max\{f_g, 0.014\}. \tag{A.12}$$

References

Akselsen, A.H., 2016. A linear stability analyses of discrete representations of the two-fluid model for stratified two-phase flow. In: Proceedings of the 9th International Conference on Multiphase Flow, May 22nd–27th ICMF-2016, Firenze, Italy.

Akselsen, A.H., 2016. Efficient Numerical Methods for Waves in One-Dimensional Two-Phase Pipe Flows. (Ph.D. thesis). NTNU.

Barnea, D., Taitel, Y., 1993. Kelvin–Helmholtz stability criteria for stratified flow: viscous versus non-viscous (inviscid) approaches. Int. J. Multiph. Flow 19 (4), 639–649. doi:10.1016/0301-9322(93)90092-9.

Barnea, D., Taitel, Y., 1994. Interfacial and structural stability of separated flow. Int. J. Multiph. Flow 20 (94), 387–414.

Bendiksen, K., Malnes, D., Moe, R., Nuland, S., 1991. The dynamic two-fluid model OLGA: theory and application. SPE Prod. Eng. 1991 6 (SPE-19451), 171–180.

Berry, R. A., Zou, L., Zhao, H., Zhang, H., Peterson, J. W., Martineau, R. C., Kadioglu, S. Y., Andrs, D., 2015. RELAP-7 Theory manual - INL/EXT-14-31366.

Biberg, D., 1999. An explicit approximation for the wetted angle in two-phase stratified pipe flow. Can. J. Chem. Eng. 77 (3), 1221–1224.

Bonizzi, M., 2003. Transient One-Dimensional Modelling of Multiphase Slug Flows. (Ph.D. thesis). Imperial College.

Butcher, J.C., 2003. Numerical Methods for Ordinary Differential Equations. John Wiley & Sons, Ltd. doi:10.1002/0470868279.

Churchill, S.W., 1977. Friction-factor equation spans all fluid flow regimes. Chem. Eng. 84 (24), 91–92.

Danielson, T.J., Bansal, K.M., Hansen, R., Leporcher, E., 2005. LEDA: the next multiphase flow performance simulator. In: BHR Group 2005 Multiphase Production Technology 12, pp. 477–492.

Drew, D.A., Passman, S.L., 1998. Theory of Multicomponents Fluids. Springer.

Evje, S., Flåtten, T., 2003. Hybrid flux-splitting schemes for a common two-fluid model. J. Comput. Phys. 192 (1), 175–210. doi:10.1016/j.jcp.2003.07.001.

Figueiredo, A.B., Baptista, R.M., de Freitas Rachid, F.B., Bodstein, G.C.R., 2017. Numerical simulation of stratified-pattern two-phase flow in gas pipelines using a two-fluid model. Int. J. Multiph. Flow 88, 30–49. doi:10.1016/j.ijmultiphaseflow.2016.09.016.

Fitt, A.D., 1989. The numerical and analytical solution of ill-posed systems of conservation laws. Appl. Math. Model. 13, 618–631.

Fullmer, W.D., Ransom, V.H., Lopez de Bertodano, M.A., 2014. Linear and nonlinear analysis of an unstable, but well-posed, one-dimensional two-fluid model for two-phase flow based on the inviscid kelvin-Helmholtz instability. Nucl. Eng. Des. 268, 173–184.

Gottlieb, S., Shu, C.-W., Tadmor, E., 2001. Strong stability-preserving high-order time discretization methods. SIAM Rev. 43 (1), 89–112. doi:10.1137/S003614450036757X.

Hairer, E., Norsett, S.P., Wanner, G., 2008. Solving Ordinary Differential Equations I. Springer. doi: 10.1007/978-3-540-78862-1.

Hendrix, M.H.W., Smith, I.E., van Zwieten, J.S.B., Sanderse, B., 2016. Comparison of numerical methods for slug capturing with the two-fluid model. In: ICMF-2016 - 9th International Conference on Multiphase Flow May 22nd - 27th 2016, Firenze, Italy.

Hirsch, C., 1994. Numerical Computation of Internal and External Flows - Volume 1, Fundamentals of Numerical Discretization. Wiley.

Holmås, H., 2008. Numerical Simulation of Waves in Two-Phase Pipe Flow Using 1D Two-Fluid Models. (Ph.D. thesis). University of Oslo.

Holmås, H., 2010. Numerical simulation of transient roll-waves in two-phase pipe flow. Chem. Eng. Sci. 65 (5), 1811–1825. doi:10.1016/j.ces.2009.11.031.

Hundsdorfer, W., Verwer, J., 2007. Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations, 33. Springer. doi: 10.1007/978-3-662-09017-6.

Issa, R.I., Kempf, M.H.W., 2003. Simulation of slug flow in horizontal and nearly horizontal pipes with the two-fluid model. Int. J. Multiph. Flow 29 (1), 69–95. doi:10.1016/S0301-9322(02)00127-1.

Johnson, G. W., 2005. A Study of Stratified Gas-Liquid Pipe Flow. (Ph.D. thesis). University of Oslo.

Kjølaas, J., De Leebeeck, A., Johansen, S.T., 2013. Simulation of hydrodynamic slug flow using the Ledaflow slug capturing model. In: Proceedings of the 16th International Conference on Multiphase Production Technology, 12–14 June, Cannes, France.

Liao, J., Mei, R., Klausner, J.F., 2008. A study on the numerical stability of the two-fluid model near ill-posedness. Int. J. Multiph. Flow 34 (11), 1067–1087. doi:10.1016/j.ijmultiphaseflow.2008.02.010.

López de Bertodano, M., Fullmer, W., Clausse, A., Ransom, V.H., 2017. Two-Fluid Model Stability, Simulation and Chaos. Springer doi:10.1007/978-3-319-44968-5.

Lopez-de-Bertodano, M., Fullmer, W., Vaidheeswaran, A., 2013. One-dimensional two-equation two-fluid model stability. Multiph. Sci. Technol. 25 (2–4), 133–167. doi:10.1615/MultScienTechn.v25.i2-4.60.

Lyczkowski, R.W., Gidaspow, D., Solbrig, C.W., Hughes, E.D., 1978. Characteristics and stability analyses of transient one-dimensional two-phase flow equations and their finite difference approximations. Nucl. Sci. Eng. 66 (3), 378–396. doi:10.13182/nse78-4.

Masella, J.M., Tran, Q.H., Ferre, D., Pauchon, C., 1998. Transient simulation of two-phase flows in pipes. Int. J. Multiph. Flow 24 (5), 739–755. doi:10.1016/S0301-9322(98)00004-4.

Montini, M., 2011. Closure Relations of the One-Dimensional Two-Fluid Model for the Simulation of Slug Flows. (Ph.D. thesis). Imperial College.

Morin, A., Flåtten, T., Munkejord, S.T., 2013. A Roe scheme for a compressible six-equation two-fluid model. Int. J. Numer. Methods Fluids 72, 478–504. doi:10.1002/fld.

Morton, K.W., Mayers, D., 2005. Numerical Solution of Partial Differential Equations. Cambridge University Press.

Prosperetti, A., Tryggvason, G., 2007. Computational Methods for Multiphase Flow. Cambridge University Press.

Ramshaw, J.D., Trapp, J.A., 1978. Characteristics, stability, and short-wavelength phenomena in two-phase flow equation systems. Nucl. Sci. Eng. 66, 93–102.

Ransom, V.H., Hicks, D.L., 1984. Hyperbolic two-pressure models for two-phase flow. J. Comput. Phys. 53, 124–151.

Ransom, V.H., Hicks, D.L., 1988. Hyperbolic two-pressure models for two-phase flow revisited. J. Comput. Phys. 75, 498–504.

Song, J., 2003. A remedy for the ill-posedness of the one-dimensional two-fluid model. Nucl. Eng. Des. 222 (1), 40–53. doi:10.1016/S0029-5493(02)00391-6.

Song, J.H., Ishii, M., 2001. One-dimensional two-fluid model with momentum flux parameters. Nucl. Eng. Des. 205, 145–158. doi:10.1016/S0029-5493(00)00351-4.

Stewart, H.B., Wendroff, B., 1984. Two-phase flow: models and methods. J. Comput. Phys. 56, 363–409.

Toumi, I., Kumbaro, A., 1996. An approximate linearized Riemann solver for a two-fluid model. J. Comput. Phys. 124, 286–300.

Wallis, G.B., 1969. One-Dimensional Two-Phase flow. McGraw-Hill.

Whitham, G.B., 1974. Linear and Nonlinear Waves. Wiley, New York.

Zou, L., Zhao, H., Zhang, H., 2015. Applications of high-resolution spatial discretization scheme and Jacobian-free Newton–Krylov method in two-phase flow problems. Ann. Nucl. Energy 83, 101–107. doi:10.1016/j.anucene.2015.04.016.

van Zwieten, J.S.B., Sanderse, B., Hendrix, M.H.W., Vuik, C., Henkes, R.A.W.M., 2015. Efficient simulation of one-dimensional two-phase flow with a new high-order Discontinuous Galerkin method, Report 15-07. Technical Report. Delft University of Technology.

Additional work and comments to Paper 4

For the simulations in this paper, a relatively simple and lightweight implementation of the two-fluid model was created in Matlab without any tracking functionality. The reason for doing this is that the Sluggit model is mainly implemented with a semi-implicit discretization, with an analytical Jacobian. It is also possible to do fully implicit simulations with Sluggit, but this functionality was developed along with this paper and is not as well tested. The simulations in this paper are focused on a fully implicit discretization in a straight pipe and a periodic domain. Consequently, it was faster and simpler to create a lightweight Matlab model for the simulations for this study.

The Matlab model is made for a single pipe with a constant inclination, with periodic boundary conditions. The model can use the same spatial and temporal schemes as described in this thesis in sections 2.6 and 2.7. In addition, the Matlab model can use the third and fourth order explicit Runge-Kutta time integration methods. Furthermore, the Matlab model utilize the concept of Automatic Differentiation [88] to calculate the Jacobian, which were found to give a significant speedup compared to calculating the Jacobian by finite differences in Matlab. The remaining details of the Matlab model is given by the model descriptions in the paper.

A significant amount of work was done when writing Paper 4, but in the end the paper ended up having too much material and some had to be left out. This material is briefly presented here (using the same notations as in the paper). Some background information from Paper 4 is also required to fully understand this following material. All of the following simulations are for the Kelvin-Helmholtz test case described in Paper 4.

A von Neumann analysis was performed, as described in the paper. This analysis was done in a general manner, without assuming anything about the spatial convection schemes. Furthermore the analysis in general in the sense that both the BDF1, BDF2 and CN methods are accounted for. The resulting matrix ($\mathbf{M}'$) for the amplification factor G is shown in equations (235) and (236) (the matrix was split in two as it became quite large). Depending on if BDF1 or BDF2 is used, and if the gas-compressibility is included, the matrix results in a

$$\mathbf{M}'(1,1) = u_l\rho_l\left(1 - e^{-\phi i} + \frac{\Psi_{mass}}{2}\left(1 - 2e^{-\phi i} + e^{-2\phi i}\right)\right)\left(\left(1-\Theta\right)G^{-1} + \Theta\right) + \rho_l\frac{\Delta s}{\Delta t}\left[a_0 + a_1G^{-1} + a_2G^{-2}\right]$$
$$\mathbf{M}'(1,2) = \alpha_l\rho_l e^{-0.5\phi i}\left(e^{\phi i} - 1\right)\left(\left(1-\Theta\right)G^{-1} + \Theta\right)$$
$$\mathbf{M}'(1,3) = 0$$
$$\mathbf{M}'(1,4) = 0$$
$$\mathbf{M}'(2,1) = -u_g\rho_g\left(1 - e^{-\phi i} + \frac{\Psi_{mass}}{2}\left(1 - 2e^{-\phi i} + e^{-2\phi i}\right)\right)\left(\left(1-\Theta\right)G^{-1} + \Theta\right) - \rho_g\frac{\Delta s}{\Delta t}\left[a_0 + a_1G^{-1} + a_2G^{-2}\right]$$
$$\mathbf{M}'(2,2) = 0$$
$$\mathbf{M}'(2,3) = \alpha_g\rho_g e^{-0.5\phi i}\left(e^{\phi i} - 1\right)\left(\left(1-\Theta\right)G^{-1} + \Theta\right)$$
$$\mathbf{M}'(2,4) = \frac{u_g\alpha_g}{c_g^2}\left(1 - e^{-\phi i} + \frac{\Psi_{mass}}{2}\left(1 - 2e^{-\phi i} + e^{-2\phi i}\right)\right)\left(\left(1-\Theta\right)G^{-1} + \Theta\right) + \frac{\alpha_g}{c_g^2}\frac{\Delta s}{\Delta t}\left[a_0 + a_1G^{-1} + a_2G^{-2}\right]$$

(234)

$$
\begin{aligned}
&\mathbf{M}'(3,1)=\left(\rho_l u_l^2\left(e^{\phi i}-1\right)-LG_{l,0}+LG_{l,1}\right)\left(\left(1-\Theta\right)G^{-1}+\Theta\right)\\
&+\frac{0.5\rho_l u_l \Delta s}{\Delta t}\left(1+e^{\phi i}\right)\left[a_0+a_1G^{-1}+a_2G^{-2}\right]+\frac{1}{A}\frac{\partial F_l}{\partial \alpha_l}e^{-0.5\phi i}\\
&\mathbf{M}'(3,2)=2\alpha_l\rho_l u_l\left(\left(1-\Theta\right)G^{-1}+\Theta\right)\left(e^{0.5\phi i}-e^{-0.5\phi i}+\frac{\Psi_{mom}}{2}\left(e^{0.5\phi i}-2e^{-0.5\phi i}+e^{-\frac{3}{2}\phi i}\right)\right)\\
&+\frac{\rho_l\alpha_l\Delta s e^{0.5\phi i}}{\Delta t}\left[a_0+a_1G^{-1}+a_2G^{-2}\right]+\frac{1}{A}\frac{\partial F_l}{\partial u_l}\\
&\mathbf{M}'(3,3)=\frac{1}{A}\frac{\partial F_l}{\partial u_g}\\
&\mathbf{M}'(3,4)=\alpha_l\left(e^{\phi i}-1\right)\left(\left(1-\Theta\right)G^{-1}+\Theta\right)+\frac{1}{A}\frac{\partial F_l}{\partial p}e^{-0.5\phi i}\\
&\mathbf{M}'(4,1)=-\left(\rho_g u_g^2\left(e^{\phi i}-1\right)-LG_{g,0}+LG_{g,1}\right)\left(\left(1-\Theta\right)G^{-1}+\Theta\right)\\
&-\frac{0.5\rho_g u_g \Delta s}{\Delta t}\left(1+e^{\phi i}\right)\left[a_0+a_1G^{-1}+a_2G^{-2}\right]+\frac{1}{A}\frac{\partial F_g}{\partial \alpha_l}e^{-0.5\phi i}\\
&\mathbf{M}'(4,2)=\frac{1}{A}\frac{\partial F_g}{\partial u_l}\\
&\mathbf{M}'(4,3)=2\alpha_g\rho_g u_g\left(\left(1-\Theta\right)G^{-1}+\Theta\right)\left(e^{0.5\phi i}-e^{-0.5\phi i}+\frac{\Psi_{mom}}{2}\left(e^{0.5\phi i}-2e^{-0.5\phi i}+e^{-\frac{3}{2}\phi i}\right)\right)\\
&+\frac{\rho_g\alpha_g\Delta s e^{0.5\phi i}}{\Delta t}\left[a_0+a_1G^{-1}+a_2G^{-2}\right]+\frac{1}{A}\frac{\partial F_g}{\partial u_g}\\
&\mathbf{M}'(4,4)=\left[\alpha_g+\frac{g}{c_g^2}\cos\left(\theta\right)\left(\frac{1}{12}\frac{D}{0.25\pi}K_5^3-0.5\alpha_g DK_4\right)+\frac{\alpha_g u_g^2}{c_g^2}\right]\left(\left(1-\Theta\right)G^{-1}+\Theta\right)\left(e^{\phi i}-1\right)\\
&+\frac{0.5u_g\alpha_g\Delta s}{c_g^2\Delta t}\left(1+e^{\phi i}\right)\left[a_0+a_1G^{-1}+a_2G^{-2}\right]+\frac{1}{A}\frac{\partial F_g}{\partial p}e^{-0.5\phi i}
\end{aligned}
\tag{235}
$$

In these expressions, the liquid is assumed incompressible while the gas is compressible. Note however that the terms in $\mathbf{M}'(4,4)$, which include the gas compressibility ($\frac{1}{c_g^2}$) typically are small compared to the first term (α_g), which arises from the pressure gradient. If the terms including the gas compressibility are ignored, a similar expression as $\mathbf{M}'(3,4)$ for the incompressible liquid is found.

The friction terms, gravitational force and driving pressure force are gathered in F_β :

$$F_\beta = \Delta s P_\beta \tau_{\beta w} \pm \Delta s P_{gl} \tau_{\beta\gamma} + g\sin(\varphi)\rho_\beta A_\beta \Delta s - \frac{dp_{body}}{ds} A_\beta \Delta s \tag{236}$$

The derivatives of F_β are determined by numerical differentiation. The linearized level gradient term is given by (minus for gas, plus for liquid):

$$\begin{aligned}
LG_{\beta,j} &= \mp g\rho_k \cos(\theta) e^{\phi ji}\left(0.5\left(DK_4 + D\alpha_\beta K_5 K_6\right) - \frac{D}{\pi} K_4 K_5^2 K_6\right) \\
K_1 &= 2\left(\frac{3\pi}{2}\right)^{\frac{1}{3}} - \pi \\
K_2 &= \left(\frac{3\pi}{2}\right)^{\frac{1}{3}} \\
K_3 &= 0.5588463977399147 \\
K_4 &= \cos\left(\alpha_l K_1 + (1-\alpha_l)^{\frac{1}{3}} K_2 - K_2 - K_2 \alpha_l^{\frac{1}{3}}\right) \\
K_5 &= \sin\left(\alpha_l K_1 + (1-\alpha_l)^{\frac{1}{3}} K_2 - K_2 - K_2 \alpha_l^{\frac{1}{3}}\right) \\
K_6 &= K_3 (1-\alpha_l)^{-\frac{2}{3}} - K_1 + K_3 \alpha_l^{-\frac{2}{3}}
\end{aligned} \tag{237}$$

The flux limiters for the mass and momentum equation Ψ_{mass} and Ψ_{mom} are described in equation (21), and takes a normalized variable *r* as input (equation (22)). Inserting a mean value plus a perturbation in equation (21), we get for *r*:

$$r = e^{\phi i} \tag{238}$$

This is independent of the time level the values for Φ are taken from (as long as they are all from the same time step). The flux limiters Ψ to be inserted in the amplification matrix $\mathbf{M}'$ are shown below for the FOU, central and NOTABLE schemes.

$$\begin{array}{ll} \text{FOU} & 0 \\ \text{CD} & e^{\phi i} \\ \text{NOTABLE} & \dfrac{3e^{2\phi i}+e^{\phi i}}{\left(1+e^{\phi i}\right)^2} \end{array} \tag{239}$$

It should also be straightforward to add more time levels to the time integration: Including the BDF3 (or BDF2.5) method, the brackets containing the a-coefficients in matrix simply reads $\left[a_0 + a_1 G^{-1} + a_2 G^{-2} + a_3 G^{-3}\right]$. This could more conveniently be expressed by the general form $\sum_{i=0}^{n} a_i G^{-i}$ for n time levels.

The amplification factor *G* obtained from the von Neumann analysis for the Kelvin-Helmholtz test case described in Paper 4 are shown in Figure 78 - Figure 80 for different spatial and temporal methods, as function of $\phi = k\Delta x = 2\pi\Delta x = \dfrac{2\pi}{N}$. The analytical amplification factor for the wave equation used in the test case is also shown.

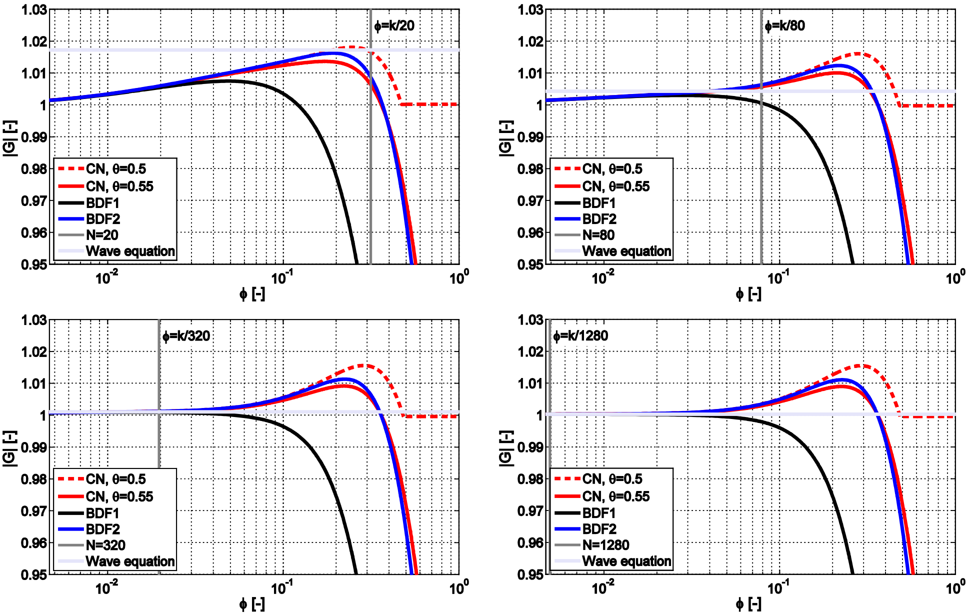


Figure 78: Amplification factor *G* for the first order upwind scheme, for different time integration methods. From top left to bottom right: N=20, 80, 320 and 1280 grid cells.

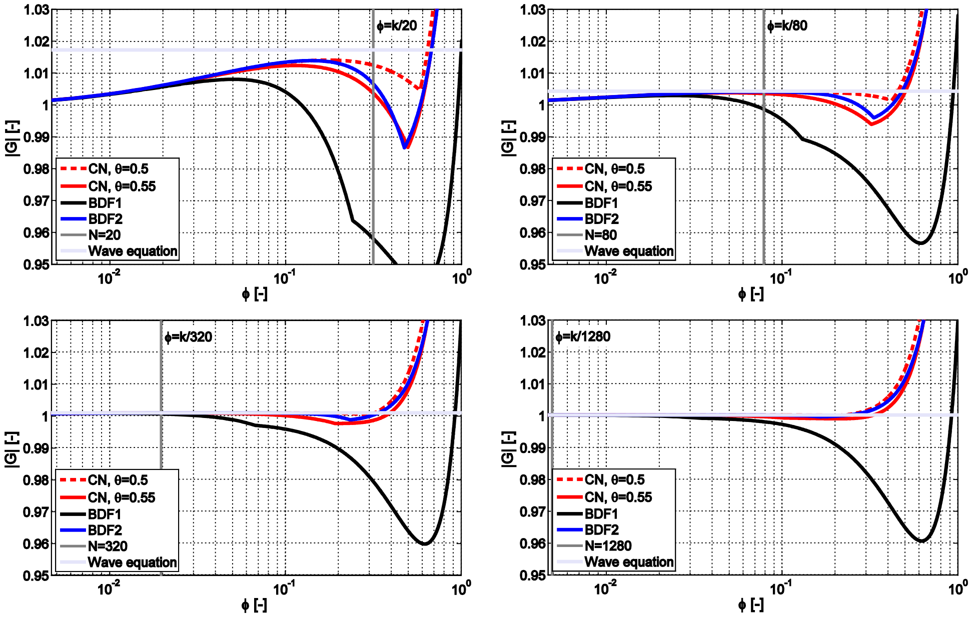


Figure 79: Amplification factor *G* for the NOTABLE scheme, for different time integration methods. From top left to bottom right: N=20, 80, 320 and 1280 grid cells.

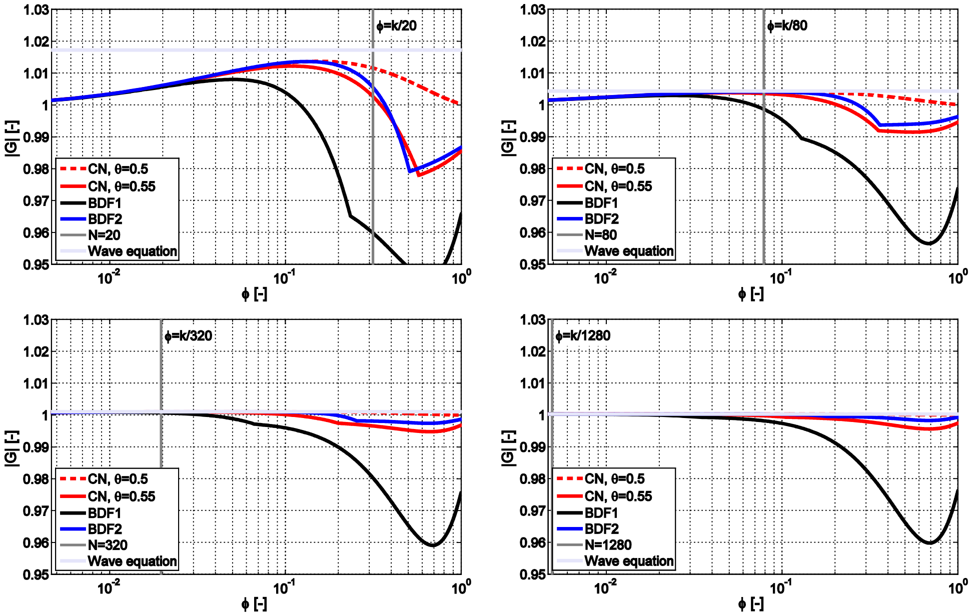


Figure 80: Amplification factor *G* for the central scheme, for different time integration methods. From top left to bottom right: N=20, 80, 320 and 1280 grid cells.

From Figure 78 - Figure 80 it can be seen that the FOU scheme is much more diffusive than the NOTABLE and central schemes, as expected. It is also evident that the BDF1 time integration method is more diffusive than the BDF2 and CN methods. Furthermore on can see that the FOU scheme is the only scheme that can give a too large amplification (see for instance N=80). This is somewhat counter-intuitive, as the FOU scheme is typically thought of as being too diffusive. This effect was however also observed in the simulations. The CN method with $\Theta = 0.5$ is the most accurate time integration method, while BDF2 and CN with $\Theta = 0.55$ are nearly identical.

The results from the von Neumann analysis were also compared with the amplification obtained from the simulations, to check that the results are consistent. This comparison is shown in Figure 81 and Figure 82. Instead of comparing with the amplification factor *G*, these plots shows the amplification factor *G*', which is the amplification per second: $G' = G^{\frac{1}{\Delta t}}$. The von Neumann analysis is seen to accurately predict the amplification factor in the simulations, as expected.

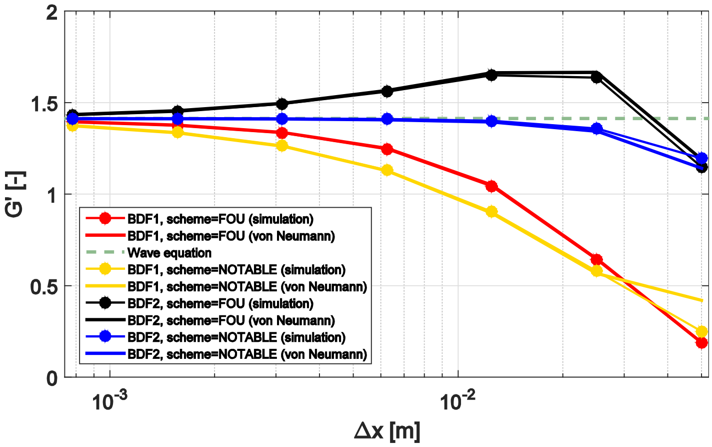


Figure 81: Comparison between the amplification factor *G*' from the von Neumann analysis for the Kelvin-Helmholtz test case and the amplification obtained from the simulations for the BDF1 and BDF2 methods.

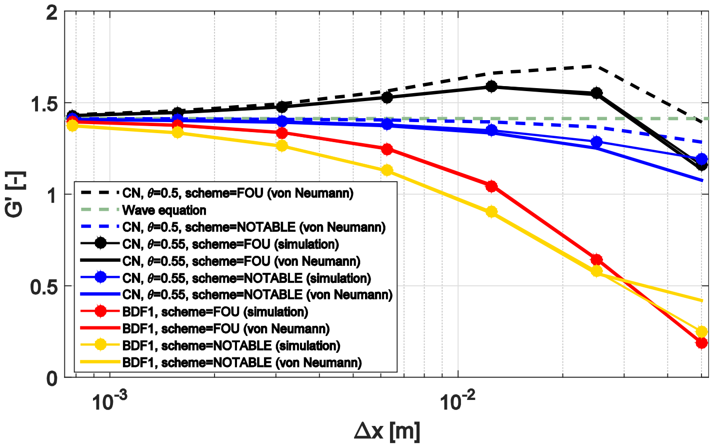


Figure 82: Comparison between the amplification factor *G'* from the von Neumann analysis for the Kelvin-Helmholtz test case and the amplification obtained from the simulations for the BDF1 and CN methods.

Similarly to what was presented in Paper 3, the time until the simulation becomes ill-posed are presented as function of time step. The CFL number is kept constant, so that plotting against the time step is similar to plotting against the grid size.

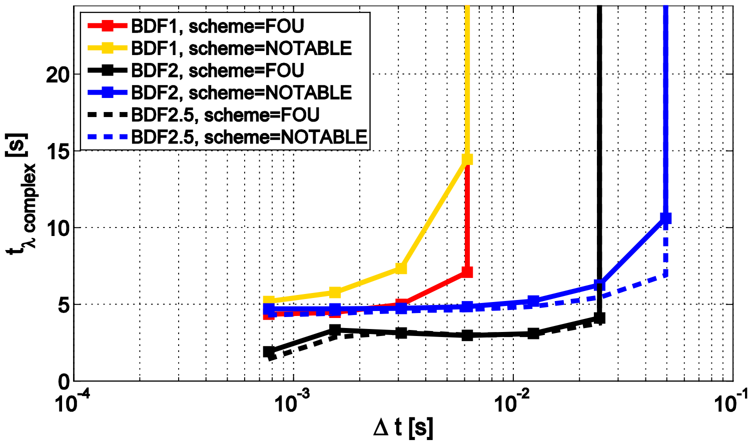


Figure 83: Time instance at which complex eigenvalues first appear, as function of time step, for different discretization methods (BDF1 and BDF2).

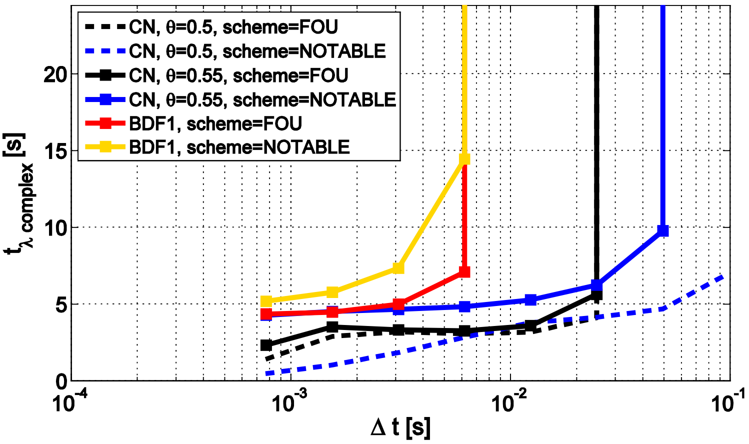


Figure 84: Time instance at which complex eigenvalues first appear, as function of time step, for different discretization methods (BDF1 and CN).

From Figure 83 and Figure 84 one can see that most of the different discretizations converge to the same solution as the grid is refined. For the first order upwind convection scheme however, it can be seen that the simulations with higher order time integration methods (BDF2 and CN) does not seem to converge to the same value as the remaining simulations. When using the BDF1 time integration method however, the simulations with the FOU scheme seems to converge to the correct value. Furthermore one can see that the CN method with $\Theta = 0.5$ shows the worst performance, and gives ill-posed results too early in the simulations. The reason for this is most likely that this method has amplification 1 for the unresolved pressure waves, so that unphysical inaccuracies that occurs in the pressure are not damped, and affects the gas density and holdup creating a growing disturbance. One can also see that by using higher order methods one will get the correct ill-posed results at a much coarser grid/time step than with the lower order methods.

A comparison was also made between the BDF2, CN ($\Theta = 0.55$) and BDF2.5 time integration methods, by plotting the holdup at location 0 m against time as shown in Figure 85 and Figure 86 (zoomed to well-posed region). These plots shows that all schemes are predicting the wave growth quite well in the well-posed regime. The CN method is slightly worse than the others in predicting the wave growth/amplitude, while it is slightly better than the others at predicting the wave speed (see the location of the peaks in the holdup signal). It can also be seen that the BDF2.5 method predicts a slightly too large wave growth. It is also worthwhile noting the differences between BDF2 and CN in the ill-posed regime: BDF2 predicts a larger amplitude than CN for the waves, which looks more similar to the analytical result (which is only valid the well-posed regime). The bottom of the wave crests from the CN method are seen to have a higher holdup value. The wave speed is also seen to change for the CN method as it enters the ill-posed regime, while the BDF2 method keeps a similar wave speed as in the well-posed region. It is however not known how the exact solution should look like in the ill-posed regime...

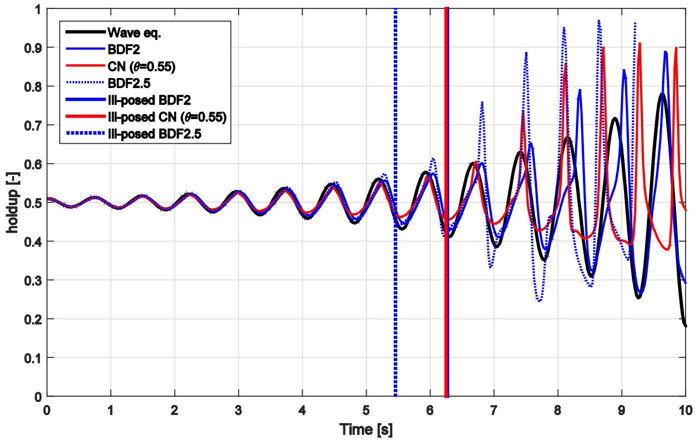


Figure 85: Holdup at location 0 m as function of time, for the BDF2, CN and BDF2.5 time integration methods. The NOTABLE scheme was used for convection of both mass and momentum, and the number of grid points are 40.

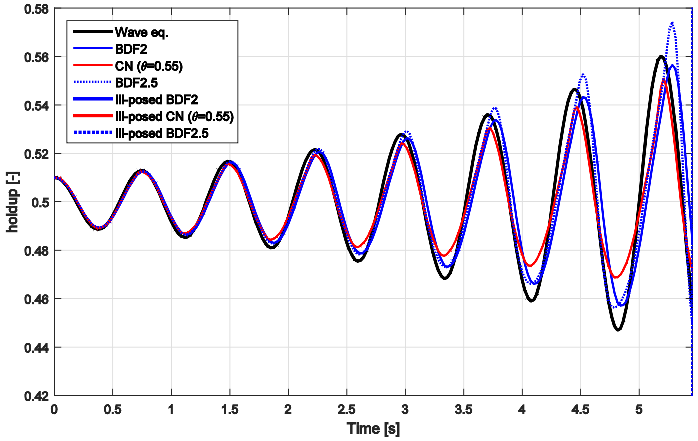


Figure 86: Holdup at location 0 m as function of time, for the BDF2, CN and BDF2.5 time integration methods showing only the time-region where the simulations are well-posed. The NOTABLE scheme was used for convection of both mass and momentum, and the number of grid points are 40.

Furthermore, many different discretizations of the two-fluid model was simulated and compared. The following properties were varied:

List 7: Discretization parameters that were varied

1. The time integration method for mass (CN with $\Theta = 0.45$ or BDF2)
2. The time integration method for momentum (CN with $\Theta = 0.45$ or BDF2)
3. Fully implicit or fully explicit momentum fluxes
4. Fully implicit or semi-explicit source terms (using one new velocity or velocity difference for wall and interfacial friction if semi-implicit). The level gradient and gravitational force were either fully implicit or explicit.
5. Using the mass flux in the momentum equation as an average of the mass equation fluxes, or using the specific mass defined at the momentum control volume end-points.

The number of grid points used was 40, and the pressure gradient, driving pressure force and the mass equation were treated as normal (that is, fully implicit for BDF2 and weighted between implicit and explicit for CN). The time step was based on both fluid velocities, which in practice means that the gas velocity controls the time step. Note that this is in opposite to most of the simulations presented in Paper 4, where the time step is based on the liquid velocity. The CFL number used was 0.8, which resulted in the fluid-CFL criterion always being obeyed (which is necessary when using explicit momentum fluxes). In total List 7 results in $2^5 = 32$ unique discretization combinations. The NOTABLE scheme was used for convection of both mass and momentum. For explicit momentum however, the FOU scheme was used as experience showed that higher order limiters could be problematic to use with explicit convection. The simulations focused on capturing the correct wave growth in the well-posed part of the simulation. If the results became ill-posed, or the holdup level became larger than 0.8 or smaller than 0.2 the simulation was stopped. The results are shown in Figure 87 - Figure 94.

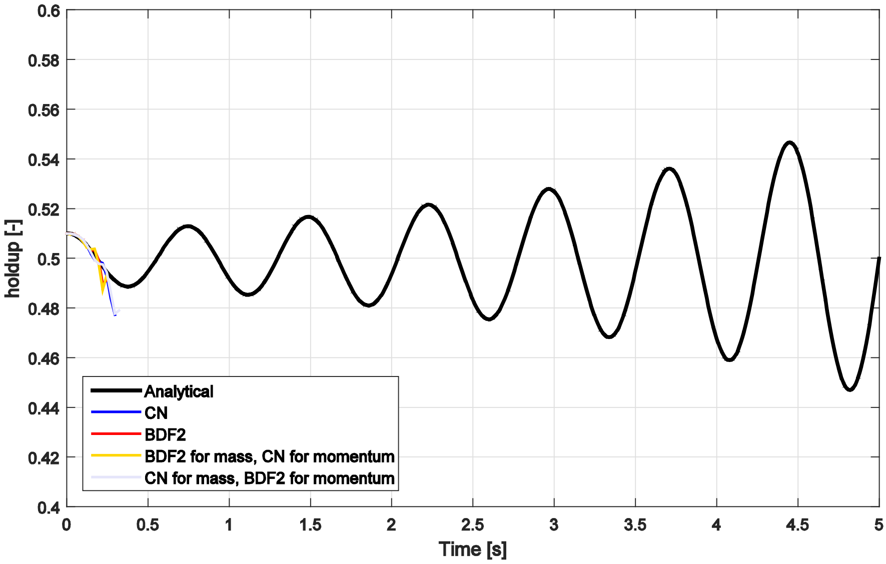


Figure 87: Results for the simulations with explicit momentum fluxes, explicit source-terms, and using the average mass flux from the mass equation in the momentum fluxes.

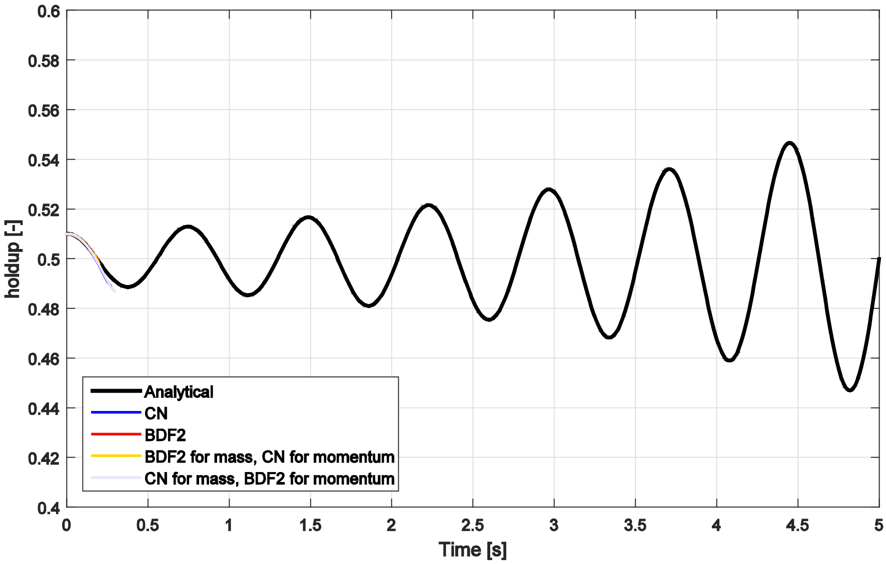


Figure 88: Results for the simulations with explicit momentum fluxes, explicit source-terms, and using the specific mass defined at the control volume end-point for the momentum fluxes.

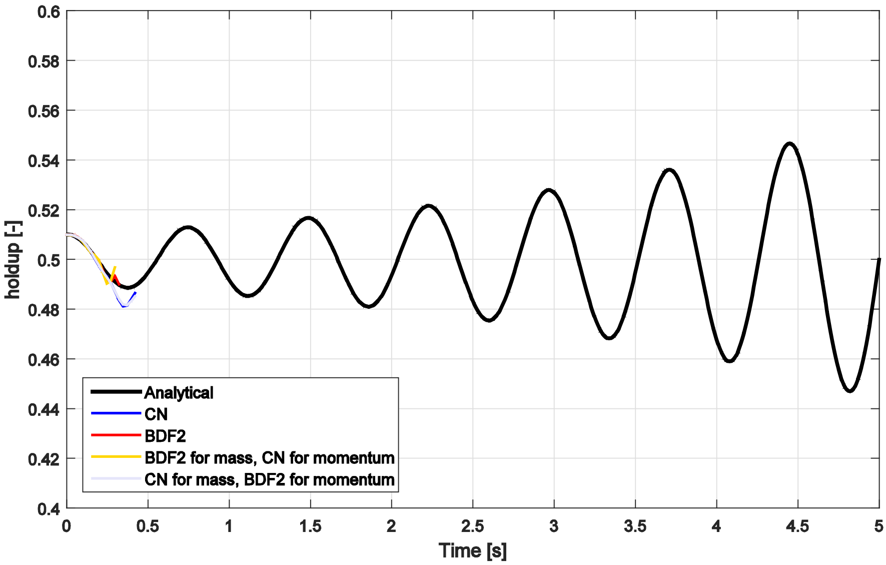


Figure 89: Results for the simulations with explicit momentum fluxes, implicit source-terms, and using the average mass flux from the mass equation in the momentum fluxes.

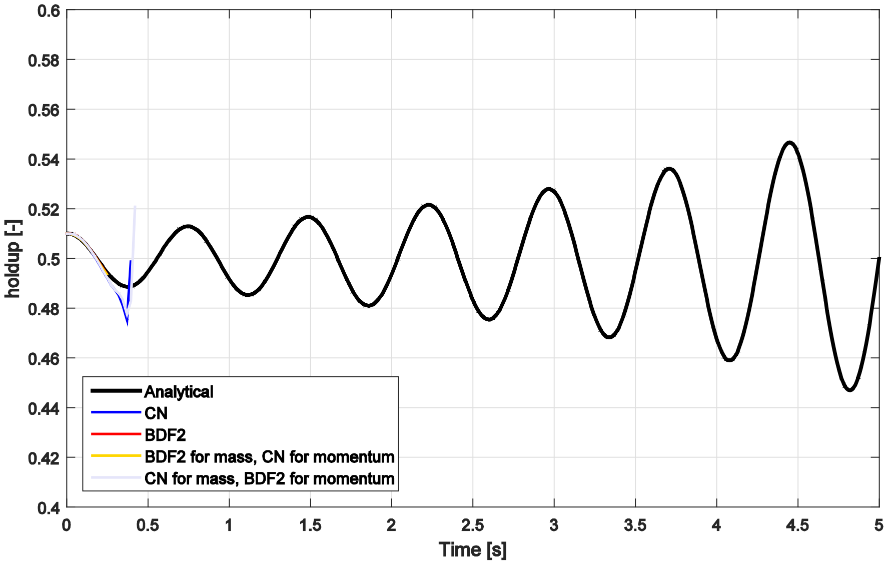


Figure 90: Results for the simulations with explicit momentum fluxes, implicit source-terms, and using the specific mass defined at the control volume end-point for the momentum fluxes.

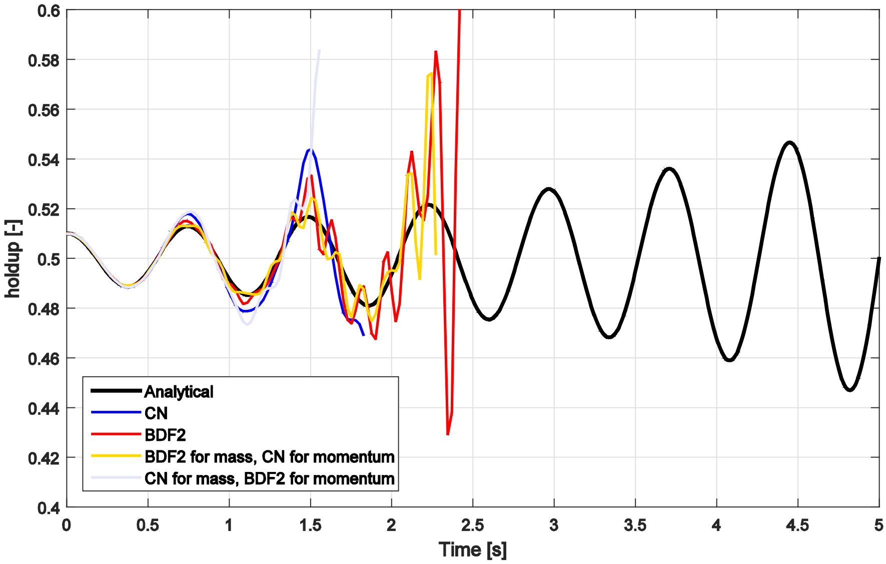


Figure 91: Results for the simulations with implicit momentum fluxes, explicit source-terms, and using the average mass flux from the mass equation in the momentum fluxes.

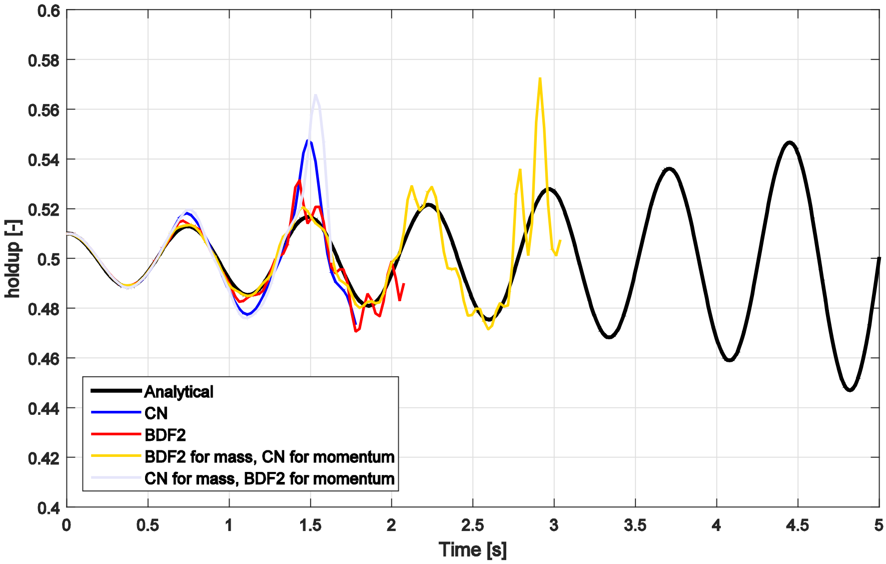


Figure 92: Results for the simulations with implicit momentum fluxes, explicit source-terms, and using the specific mass defined at the control volume end-point for the momentum fluxes.

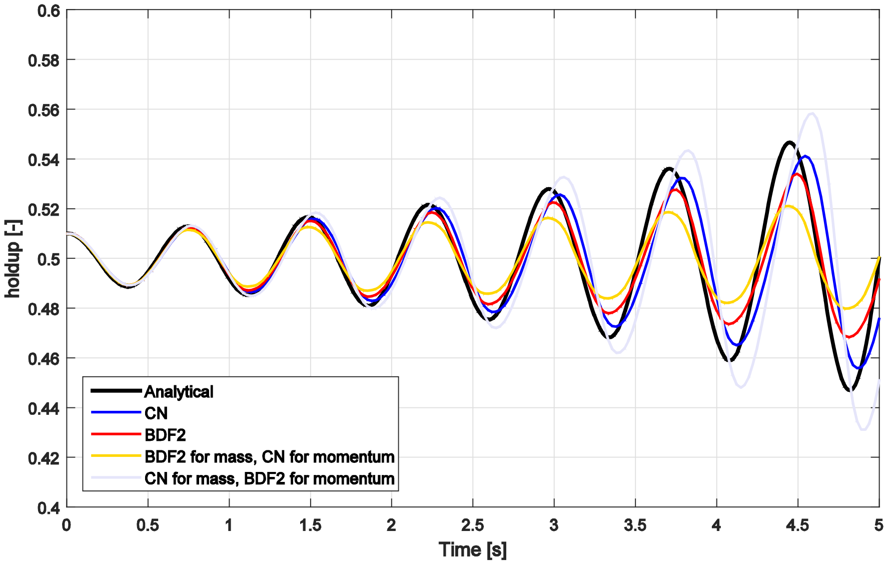


Figure 93: Results for the simulations with implicit momentum fluxes, implicit source-terms, and using the average mass flux from the mass equation in the momentum fluxes.

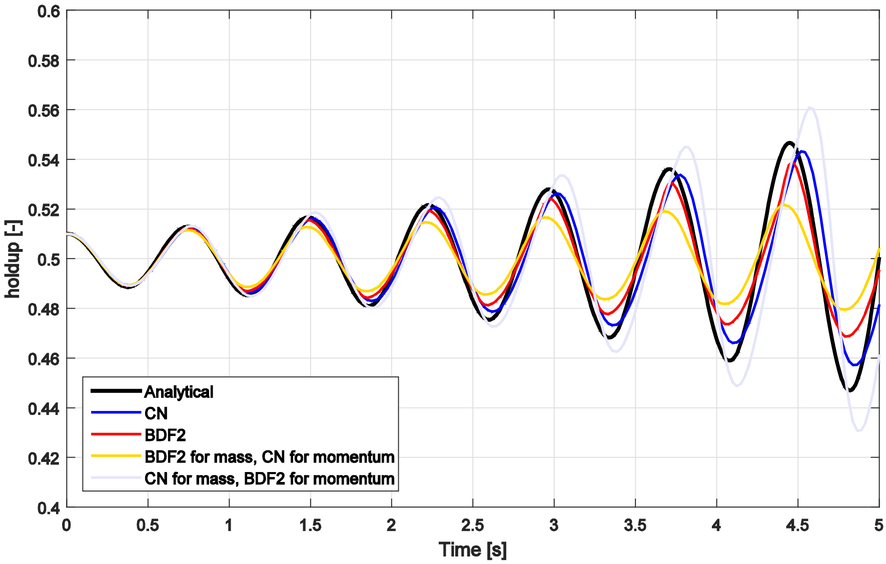


Figure 94: Results for the simulations with implicit momentum fluxes, implicit source-terms, and using the specific mass defined at the control volume end-point for the momentum fluxes.

The results from Figure 87 - Figure 94 shows that most of the simulations fail at capturing the well-posed wave growth, which indicates that those discretizations might not be a good idea to use as they do not represent an accurate and stable solution of the two-fluid model. The only discretizations that are able to capture the correct wave growth are the fully implicit simulations in Figure 93 and Figure 94. The difference between these two figures is the calculation of the mass-fluxes used in the momentum fluxes, which for this test-case only results in a negligible difference. Comparing the four different combinations of time integration methods within one figure, one can see that using BDF2 for the mass equation and CN for momentum seems to give a somewhat smaller wave growth than for the other cases. In opposite, it seems that using BDF2 for momentum and CN for mass seems to result in a too large wave growth. Using the same time integration method for both mass and momentum however seems to give the best results. Furthermore, one can see that using CN for both mass and momentum gives a slightly better prediction of the wave speed, while BDF2 is slightly better at predicting the wave growth/height. The differences between using a pure BDF2 or CN method are however quite small.

The BDF2.5 time integration scheme (taking the average of BDF2 and BDF3) was initially also used in this paper, but was left out. To obtain the coefficients for the blended BDF2.5 method for a varying time step, we need expressions for the coefficients for both the BDF2 and BDF3 methods as function of the time step ratios. The coefficients for BDF2 are easily found in the literature, and are shown in equation (31). The coefficients for the BDF3 method with a varying time step are however not typically described in the literature, but can be obtained in a similar way as for BDF2. We start by expressing the BDF3 method as:

$$\mathbf{U}^{n+1} = -\alpha_n \mathbf{U}^n - \alpha_{n-1}\mathbf{U}^{n-1} - \alpha_{n-2}\mathbf{U}^{n-2} + \beta\Delta t^n \dot{\mathbf{U}}^{n+1} \tag{240}$$

The local truncation error $\mathbf{d}^n$ is given by:

$$\mathbf{d}^n = \mathbf{U}^{n+1} - \mathbf{U}\left(t^{n+1}\right) = -\alpha_n \mathbf{U}^n - \alpha_{n-1}\mathbf{U}^{n-1} - \alpha_{n-2}\mathbf{U}^{n-2} + \beta\Delta t^n - \alpha_{n-1}\dot{\mathbf{U}}^{n+1} - \mathbf{U}\left(t^{n+1}\right) \tag{241}$$

We now do a Taylor expansion about step *n*, and for third order accuracy we require that the derivatives up to third order should vanish. This gives four equations and four unknowns, as shown in equation (243):

$$\begin{aligned}
&\mathbf{U}^n\left(-\alpha_n - \alpha_{n-1} - \alpha_{n-2} - 1\right) = 0 \\
&\dot{\mathbf{U}}^n\left(\alpha_{n-1}\Delta t^{n-1} + \alpha_{n-2}\left(\Delta t^{n-1} + \Delta t^{n-2}\right) + \beta\Delta t^n - \Delta t^n\right) = 0 \\
&\ddot{\mathbf{U}}^n\left(-\alpha_{n-1}\frac{\left(\Delta t^{n-1}\right)^2}{2} - \alpha_{n-2}\frac{\left(\Delta t^{n-1} + \Delta t^{n-2}\right)^2}{2} + \beta\left(\Delta t^n\right)^2 - \frac{\left(\Delta t^n\right)^2}{2}\right) = 0 \\
&\dddot{\mathbf{U}}^n\left(\alpha_{n-1}\frac{\left(\Delta t^{n-1}\right)^3}{6} + \alpha_{n-2}\frac{\left(\Delta t^{n-1} + \Delta t^{n-2}\right)^3}{6} + \beta\frac{\left(\Delta t^n\right)^3}{2} - \frac{\left(\Delta t^n\right)^3}{6}\right) = 0
\end{aligned} \tag{242}$$

Here the time step ratios are defined by equation (244):

$$\begin{aligned} r_{\Delta t}^{n} &= \frac{\Delta t^{n}}{\Delta t^{n-1}} = \frac{t^{n+1} - t^{n}}{t^{n} - t^{n-1}} \\ r_{\Delta t}^{n-1} &= \frac{\Delta t^{n-1}}{\Delta t^{n-2}} = \frac{t^{n} - t^{n-1}}{t^{n-1} - t^{n-2}} \end{aligned} \tag{243}$$

This gives the following coefficients:

$$\begin{aligned} \beta &= \frac{r_{\Delta t}^{n} r_{\Delta t}^{n-1}\left(r_{\Delta t}^{n} + 2\right) + r_{\Delta t}^{n} + r_{\Delta t}^{n-1} + 1}{r_{\Delta t}^{n} r_{\Delta t}^{n-1}\left(3r_{\Delta t}^{n} + 4\right) + 2r_{\Delta t}^{n} + r_{\Delta t}^{n-1} + 1} \\ \alpha_{n-2} &= \frac{-\left(\left(r_{\Delta t}^{n}\right)^{2} + 2r_{\Delta t}^{n} + 1\right)\left(r_{\Delta t}^{n}\right)^{2}\left(r_{\Delta t}^{n-1}\right)^{3}}{\left(r_{\Delta t}^{n-1} + 1\right)\left(r_{\Delta t}^{n} r_{\Delta t}^{n-1}\left(3r_{\Delta t}^{n} + 4\right) + 2r_{\Delta t}^{n} + r_{\Delta t}^{n-1} + 1\right)} \\ \alpha_{n-1} &= \frac{\left(r_{\Delta t}^{n}\right)^{2}\left(\left(r_{\Delta t}^{n} r_{\Delta t}^{n-1}\right)^{2} + \left(r_{\Delta t}^{n-1} + 1\right)\left(2r_{\Delta t}^{n-1} r_{\Delta t}^{n} + r_{\Delta t}^{n-1} + 1\right)\right)}{r_{\Delta t}^{n} r_{\Delta t}^{n-1}\left(3r_{\Delta t}^{n} + 4\right) + 2r_{\Delta t}^{n} + r_{\Delta t}^{n-1} + 1} \\ \alpha_{n} &= -\left(\alpha_{n-1} + \alpha_{n-2} + 1\right) \end{aligned} \tag{244}$$

Dividing equation (241) by β and re-arranging gives the coefficients for BDF3 of the same form as has been used for the other time integration methods in this thesis:

$$a_0 \mathbf{U}^{n+1} + a_1 \mathbf{U}^{n} + a_2 \mathbf{U}^{n-1} + a_3 \mathbf{U}^{n-2} = \beta \Delta t^{n} \dot{\mathbf{U}}^{n+1} \tag{245}$$

The *a* coefficients for BDF3 are given by equation (247).

$$a_0 = \frac{1}{\beta}, \quad a_1 = \frac{\alpha_n}{\beta}, \quad a_2 = \frac{\alpha_{n-1}}{\beta}, \quad a_3 = \frac{\alpha_{n-2}}{\beta} \tag{246}$$

For a constant time step equation (247) gives $a_0 = \frac{11}{6}$, $a_1 = -3$, $a_2 = 1.5$ and $a_3 = -\frac{1}{3}$. The coefficients for the BDF2.5 scheme for a varying time step can now also be calculated, by taking half of the BDF2 and BDF3 coefficients given by equations (31) and (247) (with $a_3 = 0$ for BDF2).