



NTNU – Trondheim
Norwegian University of
Science and Technology

Flow in Pelton turbines

Kjartan Furnes

Master of Science in Engineering and ICT

Submission date: June 2013

Supervisor: Torbjørn Kristian Nielsen, EPT

Co-supervisor: Bjørn Winther Solemslie, EPT

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

EPT-M-2013-40

MASTEROPPGAVE

for

Stud.techn. Kjartan Furnes

Våren 2013

Strømning i Pelton turbiner*Flow in Pelton turbines***Bakgrunn**

Strømningen i en Pelton turbin er påvirket av en rekke kompliserte strømningsfenomen. Disse inkluderer høye hastigheter, skarpe gradienter, fri overflate strømning, dråper og tofase strømning. Det har i det siste tiåret vært en betydelig utvikling av numerisk strømningsberegninger i Pelton turbiner. Mangel på gode og detaljerte målinger gjør at disse i hovedsak er validert utifra kvalitative kriterier (ser strømmingen korrekt ut), snarere enn sammenlikning av detaljerte målinger. For å forbedre nøyaktigheten på beregningene er det utviklet et forenklet 2D profil som muliggjør svært nøyaktige målinger i væskesjiktet.

Mål

Verifisere *Smoothed Particle Hydrodynamics* (SPH) simuleringer i DualSPHysics ved hjelp av et analytisk problem samt undersøke muligheten til å simulere strømmingen i Peltonskovler. Utgangspunktet er et tidligere Master- og prosjektarbeid gjennomført ved Vannkraftlaboratoriet. Utgangspunktet er målinger og simuleringer gjort på en forenklet 2D-modell av en Peltonskovl.

Oppgaven bearbeides ut fra følgende punkter

1. Gjennomføre enkelt parameterstudie av verifiseringsproblemet i DualSPHysics
2. Verifisering simuleringsverktøyet DualSHPysics ved hjelp av analytisk løselig problem.
3. Implementere forenklet 2D-modell av en Peltonskovl i DualSPHysics .
4. Gjennomføre og verifisere ved hjelp av resultater fra tidligere arbeid gjennomført av Klemetsen (2010).
5. Hvis tiden tillater det kan man også gjennomføre simuleringer på en full 3D-modell av en Peltonskovl.

” - ”

Senest 14 dager etter utlevering av oppgaven skal kandidaten levere/sende instituttet en detaljert fremdrift- og eventuelt forsøksplan for oppgaven til evaluering og eventuelt diskusjon med faglig ansvarlig/veiledere. Detaljer ved eventuell utførelse av dataprogrammer skal avtales nærmere i samråd med faglig ansvarlig.

Besvarelsen redigeres mest mulig som en forskningsrapport med et sammendrag både på norsk og engelsk, konklusjon, litteraturliste, innholdsfortegnelse etc. Ved utarbeidelsen av teksten skal kandidaten legge vekt på å gjøre teksten oversiktlig og velskrevet. Med henblikk på lesning av besvarelsen er det viktig at de nødvendige henvisninger for korresponderende steder i tekst, tabeller og figurer anføres på begge steder. Ved bedømmelsen legges det stor vekt på at resultatene er grundig bearbeidet, at de oppstilles tabellarisk og/eller grafisk på en oversiktlig måte, og at de er diskutert utførlig.

Alle benyttede kilder, også muntlige opplysninger, skal oppgis på fullstendig måte. For tidsskrifter og bøker oppgis forfatter, tittel, årgang, sidetall og eventuelt figurnummer.

Det forutsettes at kandidaten tar initiativ til og holder nødvendig kontakt med faglærer og veileder(e). Kandidaten skal rette seg etter de reglementer og retningslinjer som gjelder ved alle (andre) fagmiljøer som kandidaten har kontakt med gjennom sin utførelse av oppgaven, samt etter eventuelle pålegg fra Institutt for energi- og prosesssteknikk.

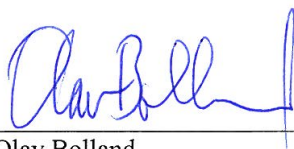
Risikovurdering av kandidatens arbeid skal gjennomføres i henhold til instituttets prosedyrer. Risikovurderingen skal dokumenteres og inngå som del av besvarelsen. Hendelser relatert til kandidatens arbeid med uheldig innvirkning på helse, miljø eller sikkerhet, skal dokumenteres og inngå som en del av besvarelsen. Hvis dokumentasjonen på risikovurderingen utgjør veldig mange sider, leveres den fulle versjonen elektronisk til veileder og et utdrag inkluderes i besvarelsen.

I henhold til "Utfyllende regler til studieforskriften for teknologistudiet/sivilingeniørstudiet" ved NTNU § 20, forbeholder instituttet seg retten til å benytte alle resultater og data til undervisnings- og forskningsformål, samt til fremtidige publikasjoner.

Besvarelsen leveres digitalt i DAIM. Et faglig sammendrag med oppgavens tittel, kandidatens navn, veileders navn, årstall, instituttnavn, og NTNUs logo og navn, leveres til instituttet som en separat pdf-fil. Etter avtale leveres besvarelse og evt. annet materiale til veileder i digitalt format.

- Arbeid i laboratorium (vannkraftlaboratoriet, strømningsteknisk, varmeteknisk)
 Feltarbeid

NTNU, Institutt for energi- og prosesssteknikk, 14. januar 2013



Olav Bolland
Instituttleder



Tobjørn K. Nielsen
Faglig ansvarlig/veileder

Medveileder: Bjørn Winther Solemslie

Preface

This master thesis was written at the hydropower Laboratory, at the Norwegian University of Science and Technology during the spring of 2013.

I'm very happy to be given the opportunity to work with an exciting and challenging task. During the last year, there have been moments when I've lost all hope. But in the end I've benefit from all the endeavours, and learned very much.

According to Nicholas Negroponte, "learning by doing, peer-to-peer teaching, and computer simulation are all part of the same equation". That is also my experience working with this thesis.

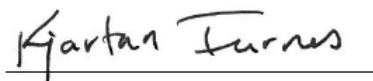
I have been standing upon the shoulders of giants. Without the great developers of DualSPHysics, there wouldn't be this great open source SPH program.

At the 8th International Spheric workshop I had the opportunity to meet the great people behind DualSPHysics and discuss some of the issues I have experience throughout the year. I learned more in a few hours then I did in weeks. These people truly known what they are talking about. I'm very grateful for the time and the response they have given me.

I'd also like to thank the scientific staff at the hydropower laboratory for encouraging and valuable guidance throughout the year, especially my co-supervisor Bjørn Winther Solemslie for having the time to discuss and reflect on issues together with me. His help and support has been invaluable.

In addition I'd like to thank all the student and the staff for a great environment at the hydropower laboratory.

Trondheim, Norway, 10.06.2013

A handwritten signature in black ink that reads "Kjartan Furnes". The signature is written in a cursive style and is positioned above a horizontal line.

Kjartan Furnes

Stud. techn.

Abstract

The flow in Pelton turbines is subsonic, turbulent, multiphase (water, air, and water vapor from cavitation), has high speeds, sharp gradients, free surface and dynamic boundary conditions. A static grid is unsuitable for modeling this mainly due to the turbine wheel and the liquid having a non-stationary relative motion.

In recent times, significant progress in CFD simulation has been made, which also is relevant for Pelton turbines. Nevertheless, it is still common to perform costly model tests to test the design of Pelton turbines. There is therefore a need to develop and implement numerical methods that allow for more realistic simulation of flows in a Pelton turbine.

In this thesis a meshfree numerical method has been studied, to investigate whether this method can be used to provide a better and more realistic simulation of flows in a Pelton turbine.

Smoothed particle hydrodynamics (SPH) is a meshfree numerical method, and has in recent years undergone considerable development. The advantage of SPH is that the method is not bound to a lattice and can better manage the free surface of a liquid motion. It uses discrete particles of fixed mass to describe fluid properties, where each particle represents mass m_i and volume V_i . SPH method approximates a function $f(x)$, using a smoothing function $W(x_i - x_j, h)$ and interpolating between the particles i and j , where the smoothing length h determines the resolution and the number of neighbors that contribute to the properties at a point. There are a number of different interpolation functions.

The purpose of this study was to investigate and assess whether the SPH based program DualSPHysics can be a good approach for simulating flows in Pelton turbines.

In this paper two test cases relevant for Pelton turbine simulations were performed, a water jet impinging normally on a fixed plate and a simple Pelton bucket geometry. The results were compared with analytical and experimental data. Comparison showed a partially good correlation between the real world and DualSPHysics.

In summary, DualSPHysics and SPH emerge as a promising tool in CFD, but this thesis shows that there is some uncertainty concerning the accuracy of the program.

Sammendrag

En Peltonturbin har subsoniske, turbulente, flerfase-strømninger (vann, luft og vandamp fra kavitasjon), høye hastigheter, skarpe gradienter, fri overflate strømning og dynamiske grensebetingelser. Et statisk grid er uegnet til dette, hovedsakelig på grunn av at turbinhullet og væsken har en ikke-stasjonær relativ bevegelse.

I den senere tid er det gjort betydelige framskritt innen CFD simulering, som også er relevant for Pelton-turbiner. Likevel er det fortsatt vanlig å utføre kostbare modell-forsøk for å prøve ut design av Peltonturbiner. Det er derfor et behov for å videreutvikle og implementere numeriske metoder som gir muligheter for mer realistiske simuleringer av strømmingene i en Peltonturbin.

I dette prosjektet har en gridfri numerisk metode blitt studert nærmere for å undersøke om denne metoden kan brukes for å gi et bedre og mer realistiske simuleringer av strømmingene i en Peltonturbin.

Smoothed particle hydrodynamics (SPH) er en gridfri numerisk metode, og har i de senere årene opplevd en omfattende utvikling. Fordelen med SPH er at metoden ikke er bundet til et gitter og kan derfor bedre håndtere den frie overflaten i en væskebevegelse. Metoden diskretiserer partikler, der hver enkel partikkel representerer masse m_i og volum V_i . SPH metoden approksimere en funksjon $f(x)$ ved hjelp av en interpolasjonsfunksjon $W(x_i - x_j, h)$ og interpolere mellom partiklene i og j , hvor interpolasjonslengden h bestemmer interpolasjonsområdet til den bestemte partikkelen. Det finnes en rekke ulike interpolasjonsfunksjoner.

Hensikten med denne studien var å undersøke og vurdere om SPH simuleringsprogramet DualSPHysics kan være en god fremgangsmåte for å simulere strømninger i Peltonturbiner.

I denne oppgaven er to tilfeller relevante for Pelton simuleringer sett på, en stråle som støter på en stasjonær plate og en enkel Pelton skovl geometri. Resultatene ble sammenlignet med analytiske og eksperimentelle data. Sammenligningenene viste en delvis god korrelasjon mellom den virkelige verden og DualSPHysics.

Kort oppsummert fremstår DualSPHysics og SPH som et spennende verktøy innenfor CFD, men denne rapporten viser også at det eksisterer en del usikkerhet rundt nøyaktigheten ved programmet.

Contents

Preface	i
Abstract	iii
Sammendrag	v
List of Figures	xiii
List of Tables	xv
Nomenclature	xvii
1 Introduction	1
2 Background	3
2.1 Problem to Solve	3
2.2 Literature Review	4
2.2.1 Numerical Analysis of a Pelton Turbine	4
2.2.2 Smoothed Particle Hydrodynamics	7
2.2.3 Jet Impinging on a plate	7
2.3 SPH Software	9
2.3.1 SPHysics and DualSPHysics	9
3 Theory	11
3.1 Pelton Turbine	11
3.1.1 Characteristics	11
3.1.2 Pelton Turbine and CFD	12
3.1.3 Flow in a Pelton Turbine	12
3.1.4 The Euler Turbomachine Equation	12
3.1.5 Velocity Components and Efficiency	12
3.1.6 Optimal Rotational Speed	14
3.1.7 Reaction Force on Pelton Turbines	15
3.2 Kinematics	16
3.2.1 Particle Paths	16

3.3	Computational Fluid Dynamics	17
3.3.1	Turbulent Flow	17
3.3.2	Validation and Verification	17
3.4	Meshfree Particle Methods	18
3.5	SPH Method	19
3.5.1	The Basics of SPH	19
3.5.2	Smoothing Length	20
3.5.3	Support Domain and Influence Domain	20
3.5.4	Integral Representation	21
3.5.5	Governing equation	21
3.5.6	Kernel Functions	23
3.5.7	Particle Approximation and Interpolation	24
3.5.8	Incompressibility	25
3.5.9	Boundary Treatment	25
3.5.10	Key Issues in SPH	26
4	DualSPHysics	29
4.1	Kernel Functions	29
4.1.1	Kernel Gradient Correction	30
4.2	Smoothing Length	30
4.3	Time stepping	31
4.4	Particle Movement	31
4.5	Boundary Conditions	32
4.6	Particle Interaction	32
4.7	Weakly Compressible SPH	33
4.8	Viscosity	34
4.8.1	Artificial Viscosity	34
4.8.2	Laminar Viscosity and Sub-Particle Scale (SPS) Turbulence	35
4.8.3	Shepard Density Filter	35
4.9	GPU	35
4.10	XML	36
4.11	Post-Processing	36
4.12	Future Improvements in DualSPHysics	36
5	Water Jet Impinging Normally on a Fixed Plate	39
5.1	Description	40
5.2	Air-Water Interaction	40
5.3	Pre-Processing	41
5.3.1	Setup	41
5.3.2	Run Script in MATLAB	42
5.4	Parameter study	42
5.4.1	Selecting of parameters	42
5.5	Validation	43
5.5.1	Stagnation pressure	43
5.5.2	Pressure Distribution	44
5.6	Post-Processing	44

5.6.1	MeasureTool	45
5.6.2	ParaView	45
5.6.3	MATLAB	45
5.7	Results	46
5.7.1	Absolute deviation	46
5.7.2	Stagnation Point Over Time	47
5.7.3	Diffusion	53
5.7.4	Repulsion	53
5.7.5	Pressure distribution	54
5.7.6	Penetration	55
5.7.7	Run time	56
6	A Simple Pelton Bucket Geometry	57
6.1	Pre-Processing	57
6.1.1	Setup	58
6.1.2	Geometry	58
6.1.3	Inlet Condition	61
6.1.4	Position Angle	61
6.2	Post-Processing	63
6.2.1	Paraview	63
6.2.2	MeasureTool	64
6.3	Validation	64
6.4	Results	64
6.4.1	Pressure at the wall	64
6.4.2	Flow Analysis	66
7	Discussion	69
7.1	Accuracy	69
7.2	Consistency	69
7.3	Pressure Analysis	70
7.4	Flow Analysis	71
7.5	Incompressibility	71
7.6	Choosing Parameters and Functions	71
7.6.1	Artificial Speed of Sound	71
7.7	Viscosity	72
7.8	Boundary Treatment	72
8	Conclusion	73
9	Further Work	75
9.1	Further Comparison to Previous Studies	75
9.2	Inlet Conditions	75
9.3	Parameters in DualSPHysics	76
9.4	Equipment for Counteracting Effect of Repulsion	76
9.4.1	Flow Analysis	76

A	DualSPPhysics XML file example	81
B	Matlab Script	83
B.1	Run DualSPPhysics Script	83
B.2	Stagnation Pressure Analysis	87
B.3	Stagnation Pressure Plot	89
B.4	Inlet Condition	91
B.5	Pressure Obtained at Bucket Wall	93
B.6	Post-process of a plane with points	95
C	Results	97
D	Table of simulation number	99

List of Figures

1.0.1 SPH showcase on a Pelton turbine from the NextMUSE project. . .	2
2.2.1 Time and azimuthal averaged pressure coefficient profile on a solid plate, comparison of old and new boundary treatment with experiments and a VOF method [39].	5
2.2.2 Hydraulic torque predicted by the SPH-ALE method and CFX [38].	6
2.2.3 The jet is at steady-state, the particle are coloured with pressure levels [40].	7
2.2.4 Time history of analytical and calculated pressure at stagnation point [28].	8
2.3.5 SHP showcase on a Pelton turbine from DualSPHysics.	9
3.1.1 Pelton turbine [51]: (a) side view of wheel and jet; (b) top view of bucket; (c) typical velocity diagram	13
3.1.2 (a) Timeline is generated by drawing a line through adjacent particles in flow at any instant of time. (b) Pathline is the line traced by a given particle. (c) Streakline concentrates on fluid particles that have gone through a fixed station or point. At some instant of time the position of all these particles are marked and a line is drawn through them [4].	17
3.5.3 The concept of support domain. The circle represents the support domain of the corresponding particles [32].	20
3.5.4 The concept of influence domain. The solid lines represent the influence domain of the corresponding particles [32].	21
3.5.5 Particle approximations using particles within the support domain[S] of the smoothing function[$W(x_i - x_j, h)$] for particle i. The support domain is circular with a radius of κh [32].	24
3.5.6 Interior fluid particles at the bound.	26
4.6.1 The circle illustrates the number of particles in the linked list with the Cubic Spline kernel.	33
4.12.2 Case with a non viscous jet impinging on a flat plate, $\theta = 30^\circ$ [10] .	37
5.0.1 Impinging Jet Configuration [26]	39
5.2.2 Definition Sketch [8]	40

5.2.3	Jet Impingement Region [36]	41
5.5.4	Radial pressure distribution [6]	44
5.6.5	A water jet impinging a plate plotted in ParaView.	45
5.7.6	Absolute deviation over time	46
5.7.7	Absolute deviation over time	47
5.7.8	Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 2$ and the dashed line is the theoretical pressure.	48
5.7.9	Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 5$ and the dashed line is the theoretical pressure.	48
5.7.10	Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 10$ and the dashed line is the theoretical pressure.	49
5.7.11	Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 20$ and the dashed line is the theoretical pressure.	49
5.7.12	Dimensionless stagnation pressure over time, $c_0/U_0 \simeq 5$ and different particle space.	50
5.7.13	Dimensionless stagnation pressure over time, $c_0/U_0 \simeq 5$ and different viscosity coefficient.	50
5.7.14	Dimensionless stagnation pressure over time, $c_0/U_0 \simeq 5$ and different number of steps to update the density at the plate.	51
5.7.15	Jet impinging a plate. Evaluated with MeasureTool and mean values around the stagnation point from ParaView (18 particles and 50 particles).	52
5.7.16	Enlarged view of the particles distribution close to the stagnation point obtained without density diffusion correction. Dimensionless pressure values are shown.	53
5.7.17	Repulsion at the plate.	53
5.7.18	Illustration of the surface pressure at different time steps. The colored contours describe the pressure, a high pressure is indicated by the red color.	54
5.7.19	The pressure distribution along the plate.	54
5.7.20	Percent of water particles that penetrated the plate at $tU/L = 2$.	55
5.7.21		56
6.1.1	Schematic drawing of the "bucket", three cylinder shapes at three different origins [29].	59
6.1.2	Point adjustment for the geometry.	59
6.1.3	The final geometry for SPH simulations.	60
6.1.4	Example code from XML file.	60
6.1.5	The generated inlet velocity profile, 100 cylinder walls were used. Plotted in ParaView.	61
6.1.6	Cross section of the flow domain [29].	62
6.2.7	Isosurface of water jet impinging the bucket and processed in Blender.	63
6.2.8	The clip function with the clip type sphere is used.	64
6.4.9	Comparison between experimental and numerical results for wall pressure along the centerline.	65
6.4.10	Pressure distribution on the bucket	66

6.4.11 Outflow	67
6.4.12 Water film thickness over the bucket, each line represent a time step.	68
7.2.1 Problem domain with SPH particle interactions across some segmenting plane. [25]	70
C.0.1 Dimensionless stagnation pressure over time.	97
C.0.2 Dimensionless stagnation pressure over time.	98
C.0.3 Dimensionless stagnation pressure over time.	98

List of Tables

5.3.1 Setup variables	42
5.4.2 Simulated parameters	42
6.1.1 Setup variables	58
6.4.2 Outflow angle	67
D.0.1 Simulation parameters	99

Nomenclature

α	empirical viscosity coefficient	
β	exit angle	<i>deg</i>
η	impulse turbine efficiency	
κ	coefficient to calculate the smoothing length (h)	
Ω	control volume	
ω	angular velocity	<i>rad/s</i>
ϕ	peripheral velocity factor	<i>kg/m³</i>
Π	Monaghan's artificial viscosity	
ρ	density	<i>kg/m³</i>
σ	surface tension coefficient	<i>N/m</i>
θ	outflow angle	<i>deg</i>
ν	kinematic viscosity	<i>m²/s</i>
ε	parameter in XSPH approach	
C_{coef}	speed of sound coefficient	
d_j	diameter of water jet	<i>m</i>
D_r	diameter of runner	<i>m</i>
r_{ij}	distance between particle i and j	<i>m</i>
u_r	rotor-tip velocity	<i>m/s</i>

V_j	water jet velocity	m/s
V_t	absolute circumferential velocity	m/s
B	constant in Tait's equation	
c	speed of sound	m/s
Cp	pressure coefficient	
Cv	velocity coefficient	
dp	particle spacing	m
F	force	N
Fr	Froude number	
g	acceleration of gravity	m/s^2
H	hydraulic head	m
h	smoothing length	m
M	torque	$N \cdot m$
Ma	Mach number	
n	revolutions per minute	rpm
Q	volume flow	m^3/s
Re	Reynolds number	
u	peripheral velocity,	m/s
v	absolute velocity	m/s
W	smoothing function	
w	relative velocity,	m/s

Abbreviation

ALE	Arbitrary Lagrangian Eulerian method
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit

CUDA Compute Unified Device Architecture
DBS Dynamic Boundary Condition
FDM Finite-Difference Methods
FEM Finite Element Method
FVM Finite-Volume Method
GPU Central Processing Unit
GPU Graphics Processing Unit
MPM Meshfree Particle Methods
NTNU Norwegian University of Science and Technology
PDE Partial Differential Equation
SPH Smoothed Particle Hydrodynamics method
SPHERIC SPH European Research Interest Community
SPS Sub-Particle Scale
WCSPH Weakly compressible SPH
XML Extensible Markup Language

Chapter 1

Introduction

Hydropower is vital in the generation of the worlds electricity, and provides around 19 % of the worlds electricity [35]. Hydropower is considered a clean energy source, and there is a large interest in achieving better efficiency designs.

The Pelton turbine is a hydraulic impulse machine developed in 1889 by Lester Allan Pelton. The buckets of the turbine wheel are impacted by water jets, and the kinetic energy of the water is transformed into mechanical energy, and finally electrical power. In contrast with most of hydraulic turbines, the flow in a Pelton turbine is only confined in the penstock, the distributor and the nozzles. The flow of water jet on a Pelton bucket is free surface flow [39].

Flows in a Pelton turbine involve various phenomena. The main challenge with conventional grid-based numerical methods such as *Finite Volume* (FV) and the *Volume of Fluids* (VOF) is the design of of the computational mesh. Especially the the numerical diffusion of the free surface is hard to remove. There are also difficulties with stability when choosing a small spacing between static and rotating parts [39].

There are several software solutions for fluid flow simulation, specialized for mechanical interaction and multiphase flows. In recent years, significant progress has been made in this type of simulation, which will also be relevant for Pelton turbines. It is still common to perform costly model experiments to test the design of Pelton turbines. There is therefore a need to develop and implement numerical methods that allow for more realistic simulation of the free surface flows encountered in Pelton turbines.

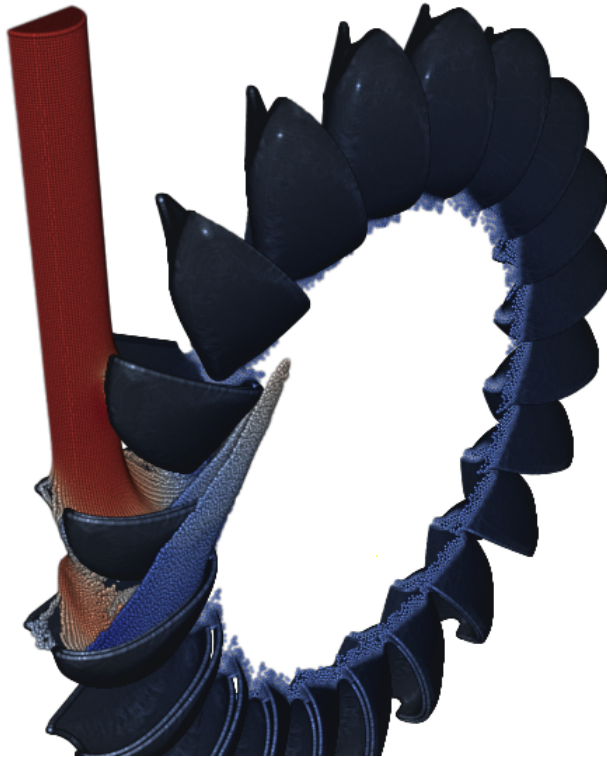


Figure 1.0.1: SPH showcase on a Pelton turbine from the NextMUSE project.

Limitations presented above explain why it is difficult to simulate a Pelton turbine with Eulerian (also called mesh-based) numerical methods. A Lagrangian method has better capacities to track interfaces properly. *The Smoothed Particle Hydrodynamics* (SPH) method is a Lagrangian numerical method and was originally developed for astrophysical simulations and has been extended to simulate fluid dynamics. SPH use calculation points which are free to move together with the fluid and do not need complex techniques like adaptive mesh required by Eulerian methods. It has advantages in that it is not affected by the numerical diffusion and can simulate a large deformation of the free surface flows relatively easily.

Numerous computational fluid dynamics (CFD) analyses on the flow in Pelton buckets have previously been carried out.

Chapter 2

Background

In this chapter a framework for this paper is presented, together with a description of the relevant literature on the topics for this study.

2.1 Problem to Solve

The objective of this study is to validate and verifying the numerical simulation software DualSPHysics. DualSPHysics is based on the SPH numerical method. The main problem with implemented SPH software is high focus on the fluid kinematics instead of the prediction of the pressure field. In general the flow speeds look quite good, but checking the distribution of pressure the situation is different [40]. Often in simulations large random pressure oscillations occurs due to to numerical high frequencies acoustic signal [9].

Another problem with SPH implementations is the long computational runtime, meaning that SPH is rarely applied to large domains. However, GPU acceleration and parallel computing appear to partially solve this problem. DualSPHysics is implemented to carry out simulations on the CPU and GPU respectively, and optimize parallelism. Crespo et al. [11], simulated more than one million particles with DualSPHysics on a single GPU card, and the simulation was 64 times faster than on a single-core CPU.

The aim of this thesis is to produce a quantity of numerical data on cases similar to that of the flow in a Pelton bucket and compare them to experimental data or analytical data. In order to assess the ability to simulate flows in Pelton buckets with the program DualSPHysics.

In short, to validate the test cases in this thesis with DualSPHysics the comparison between computational results and the experimental or analytical data have to be sufficiently accurate.

2.2 Literature Review

Computational fluid dynamics (CFD) has been a hot topic of research these last forty years, and several books have been published on this topic. Meshfree numerical methods have been much less investigated than mesh-based techniques, although they can bring clear advantages on specific flow configurations.

The research of the SPH method is promising, however, the implementation of SPH are still not satisfactory due to performance and accuracy, and further development and improvements are needed. Today, the accuracy of the torque prediction in Pelton turbine simulations is still not adequate, mainly due the development of complex secondary flows [46] [2].

To the author's knowledge and according to SPHERIC there exist only four textbooks on the SPH numerical method, where the book *Fluid Mechanics and the SPH Method* (2012) by D. Violeau is superior from an engineers point of view in the author's opinion.

2.2.1 Numerical Analysis of a Pelton Turbine

Previous research on Pelton turbines consist mainly of analytical, numerical and experimental studies [46]. This section will briefly highlight the relevant results from numerical studies of interest to this thesis.

Nakanishi, Y., Kubota, T., Shin T. (2002), *Numerical Simulation of Flows on Pelton Buckets by Particle Method* [44], a numerical simulation based on the Moving-Particle Semi-Implicit Method of flows impinging on a stationary and rotating flat plate. The force exerted by the jet on the plate and the flow rates of the divided branches were in good qualitative agreement with corresponding theoretical result.

Perrig, A. (2007), *Hydrodynamics of the Free Surface Flow in Pelton Turbine Buckets* [46], Perrig studied the flow in the buckets with four experimental and numerical approaches, unsteady onboard wall pressure measurement, high-speed flow visualizations, onboard water film thickness measurements and CFD simulations.

Perrig states that the main drawback associated with the classic CFD approaches is the need for refined meshes in the domain, as this dramatically increases the computational cost, accurate numerical simulations of a full-scale facility appears highly difficult.

Marongiu, J.C., Leboeuf, F. and Parkinson, E. (2007), *Numerical Simulation of the Flow in a Pelton Turbine Using the Meshless Method Smoothed Particle Hydrodynamics: a New Simple Solid Boundary Treatment* [39], a parallel numerical simulation with SPH was conducted, with a static bucket. The result was promising, but the pressure prediction on sharp edges was

not satisfying. The boundary treatment was a key issue. A new mathematically model for the boundary treatment was presented.

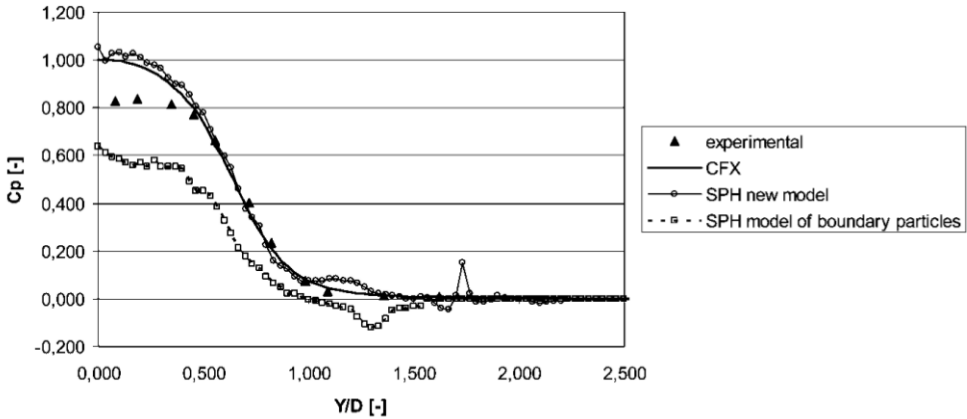


Figure 2.2.1: Time and azimuthal averaged pressure coefficient profile on a solid plate, comparison of old and new boundary treatment with experiments and a VOF method [39].

Koukouvini, P. K., Anagnostopoulos J. S., Papantoni, D. E. (2009), *Flow Modelling in the Injector of a Pelton Turbine* [30], attempted to use SPH for modelling the fluid flow in the injector of the Pelton type. Implemented the standard SPH along with a special technique, where only one symmetric part of the fluid domain is solved and the implementation was tested at the case of a jet impinging on a plate. It proved to be a good prediction of the flow and the pressure profile. It was also substantially faster than the standard SPH implementation.

Marongiu, J.C., Leboeuf F., Caro, J. and Parkinson, E. (2010), *Free Surface Flows Simulations in Pelton Turbines Using an Hybrid SPH-ALE Method* [38], an Arbitrary Lagrange Euler(ALE) and the meshless numerical method Smoothed Particle Hydrodynamics(SPH) was used to simulate the free surface flows encountered in Pelton turbines. It was shown that it is possible to accomplish good accuracy with use of the SPH method together with the ALE description. Nevertheless, the accuracy of the method is still dependent on the relatively simple SPH integration scheme.

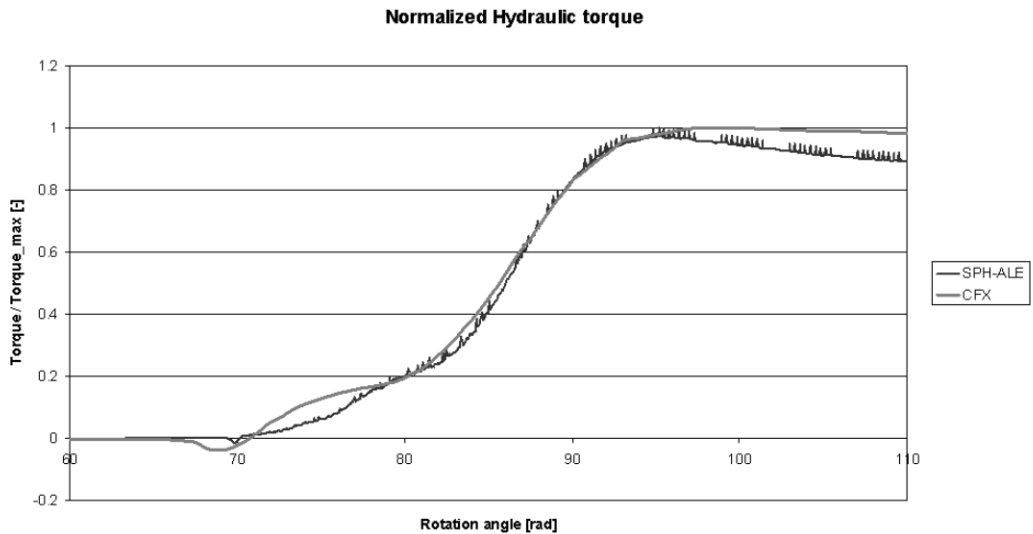


Figure 2.2.2: Hydraulic torque predicted by the SPH-ALE method and CFX [38].

Klemetsen, L. (2010), *An Experimental and Numerical Study of the Free Surface Pelton Bucket Flow* [29], the free flow surface flow trough a static Pelton turbine bucket was numerically simulated with two commercial CFD-solvers; CFX and Fluent. The numerical simulations were compared to experimental data, and a good consistency was found. The comparing suggested that the jet might not be 2-dimensional and axis-symmetrical in reality, based on discrepancies between calculated and measured wall pressure.

Barstad, L. F. (2012), *CFD-analysis of a Pelton Turbine* [5], developed and validated a numerical model for the torque applied to a non-stationary Pelton bucket. The model was developed in CFX and based on a model turbine supplied by the turbine producer DynaVec. A comparison of experimental and simulated results showed a torque over-prediction of approximately 1.5 %.

Anagnostopoulos, J. S., Papantonis, D. E. (2012), *A Fast Lagrangian Simulation Method for Flow Analysis and Runner Design in Pelton Turbines* [2], an algorithm based on the Lagrangian approach and unsteady free-surface flow during the jet-bucket interaction is simulated. Terms are introduced into the particle motion equation to account for various hydraulic losses and the flow spreading, and than to perform numerical design optimization of the bucket shape. The results showed a remarkably higher hydraulic efficiency in the entire load range.

2.2.2 Smoothed Particle Hydrodynamics

The SPH method has in recent years undergone a development and some of the initial problems with the method have been solved, and there is an increasing interest in the SPH method. The method is still little used compared with other meshfree methods [32].

The European research community for SPH, SPHERIC has initiated projects like SPHysics and NextMuse. SPHERIC [1] is a special interest group supported by ERCOFTAC. SPHERIC has given increased focus on SPH in recent years. Particularly in research and development environments in Europe. Most validations of the method is performed through Spheric [1].

In literature it is well known that the SPH method can be affected by numerical noise on the pressure field when dealing with liquids, and is today one of the main challenges faced in implementing the method [40].

2.2.3 Jet Impinging on a plate

Molteni, D., Colagrossi, A. (2009), *A Simple Procedure to Improve the Pressure Evaluation in Hydrodynamic Context using SPH* [40], implement procedures to improve the computation of the pressure distribution in the dynamics of liquids. Based on the use of a density diffusion term in the equation for the mass conservation. The study showed promising results, but a more general mathematical investigation is required. The new scheme was tested with a set of examples, i.a. a jet impinging on a flat plate.

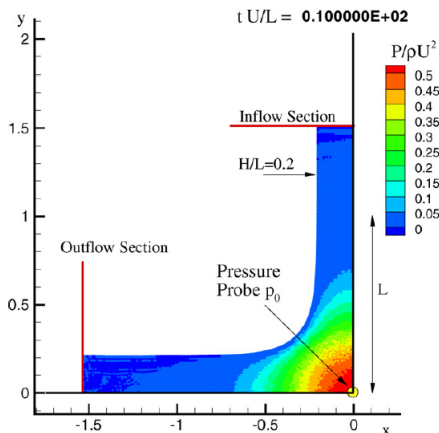


Figure 2.2.3: The jet is at steady-state, the particle are coloured with pressure levels [40].

Khayyer, A., Gotoh, H. (2011), *Enhancement of Stability and Accuracy of the Moving Particle Semi-Implicit Method* [28], investigates the performance and stability of MPS method in simulation of general hydrodynamic problems. One of the problems is a liquid jet impinging on a flat plate. After the impact of the impinging jet, the pressure approaches the theoretical one corresponding to a steady state flow regime.

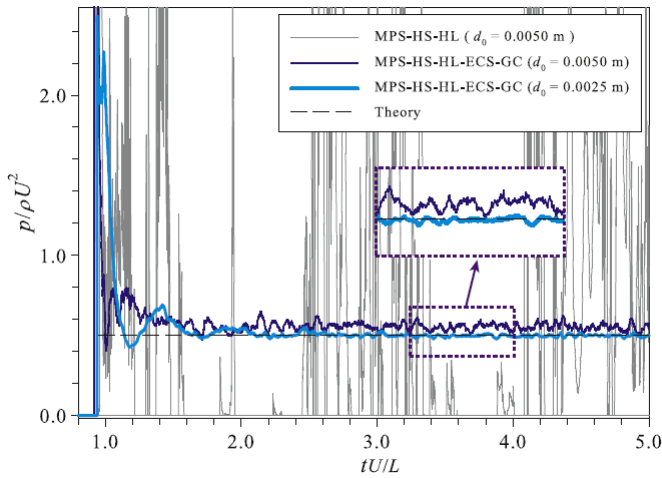


Figure 2.2.4: Time history of analytical and calculated pressure at stagnation point [28].

2.3 SPH Software

In this study the SPH software DualSPHysics (chapter 4) has been used. DualSPHysics is a set of C++ and CUDA codes. CUDA code can run on massively parallel, this includes several GPU types as well as non-GPU hardware. Massively parallel hardware can run a significantly larger number of operations per second than the CPU, at a fairly similar financial cost, yielding performance improvements of 50 times or more in situations that allow it. DualSPHysics is open-source code; free flow of information and flexibility are reasons why this is an excellent idea. This allows users to modify the program to fit their needs and gain a greater understanding of the program.

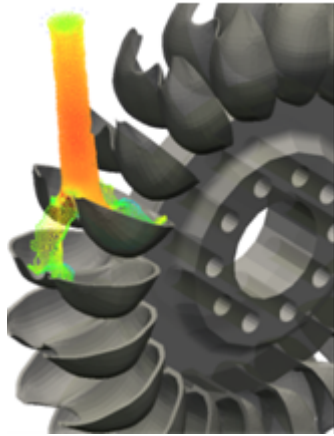


Figure 2.3.5: SHP showcase on a Pelton turbine from DualSPHysics.

2.3.1 SPHysics and DualSPHysics

The parallel version DualSPHysics of the program SPHysics has not been much documented or validated. The best documented case with DualSPHysics is the validation of the SPHERIC benchmark test case 2 [16]. However, the SPHysics Fortran code has been validated for different problems [14].

In two papers the free-surface fluid solver SPHysics is presented by the developers [20] [21]. The first paper provides a description of the formulations implemented in the code. The second paper analyses the efficiency of SPHysics by means of several test cases. Numerical results are compared to experimental data for several cases.

In 2008 the Dane Mads-Peter Hansen performed his master thesis with SPHysics. He used SPHysics to study wave flume [24].

Chapter 3

Theory

In this chapter, relevant theory behind this thesis is presented.

3.1 Pelton Turbine

The Pelton turbine is a hydraulic impulse machine developed in 1889 by Lester Allan Pelton. The buckets are impacted by water jets, and the kinetic energy of the water is transformed into mechanical energy, and finally electrical power. Design of the bucket is an important issue for the turbine efficiency.

3.1.1 Characteristics

The Pelton turbine, which is an impulse turbine, is composed of a wheel with buckets and a nozzle. Pelton turbines are usually used with high heads and with relatively low-volume flows. The runner is usually composed of 18 to 26 buckets. The water is accelerated through the nozzle and the pressure energy is converted into velocity energy at the outlet of the nozzle. The kinetic energy of the water is converted into rotational energy by deflecting the water jets flow in the impeller, which generates mechanical energy on the shaft. [7].

In an impulse turbine, all the available energy of water is converted into kinetic energy or velocity head by passing it through a contracting nozzle provided at the end of penstock. The water coming out of the nozzle has a circular cross-section. The water jet moves freely in air and impinges on a series of buckets of the runner, thus causing it to revolve. The performance of the turbine depends upon many factors and one of them is the shape of jet striking the turbine bucket which depends upon the shape of the nozzle.

3.1.2 Pelton Turbine and CFD

Earlier model testing was the only method available for assessing performance of impulse turbines for different nozzle and bucket shapes. But this approach is time consuming, costly and did not provide detailed flow behavior. With the improvements in the field of computers and advancement in numerical techniques, detailed flow analysis in given flow domain can be obtained for its design optimization [23].

Pelton turbines involve a number of special flow characteristics which are difficult to simulate. The jet-to-bucket interaction is fully transient and depends on the geometry of the moving buckets. Another challenging is the multiphase system of air and water that governs the formation of the free jet and the flow through the buckets. [46]

3.1.3 Flow in a Pelton Turbine

The Pelton turbine combines four types of flow, steady-state flow in the piping system and injector, free water jets, 3D unsteady free surface flow in the buckets and dispersed two phase flow in the casing [46].

3.1.4 The Euler Turbomachine Equation

The angular momentum equation theorem applied to turbomachinery gives the applied torque T:

$$T = \rho Q(r_2 V_{t2} - r_1 V_{t1}) \quad (3.1.1)$$

where V_t are the absolute circumferential velocity components of the flow. Base on 3.1.1 Euler developed the head pressure equation created by the impeller:

$$H = \frac{1}{g}(u_{r2} V_{t2} - u_{r1} V_{t1}) \quad (3.1.2)$$

This is the Euler turbomachine equation, and shows that the torque, power and ideal head are functions only of rotor-tip velocities(u_r) and the absolute fluid tangential velocities(V_t) independent of the axial velocities through the machine [51].

3.1.5 Velocity Components and Efficiency

The overall efficiency will be less than maximum theoretical efficiency because of friction in bearing, windage, backsplashing and nonuniform bucket flow. The losses due to bearing friction and windage increase rapidly with speed. An overall efficiency of 85-90 percent may usually be obtained in large machines. To obtain

high values of wheel efficiency, the buckets should have smooth surface and be properly designed [51].

Assuming there are no losses, all of the hydraulic potential energy is converted into kinetic energy, the theoretical maximum jet velocity is then:

$$\begin{aligned} E_k &= E_p \\ mV_j^2 &= mgH \\ V_j &= \sqrt{2gH} \end{aligned} \quad (3.1.3)$$

In a single Pelton bucket, where u is the impeller velocity and β is the exit angle of the jet. For a Pelton wheel where buckets keep entering the jet and capture all the flow, the mass flow would be $\rho Q = \rho A_j V_j$, from the Euler turbomachine equation [51].

The absolute exit and inlet tangential bucket velocity is equal ($u_1 = u_2 = u$). The turbine power relation is then [51]:

$$\begin{aligned} P &= \rho Q(u_1 V_{t1} - u_2 V_{t2}) = \rho Q u V_j - u[u + (V_j - u)\cos\beta] \\ &= \rho Q u (V_j - u)(1 - \cos\beta) \end{aligned} \quad (3.1.4)$$

where $u = 2\pi nr$ is the bucket linear velocity and r is the pitch radius, or distance to the jet centerline.

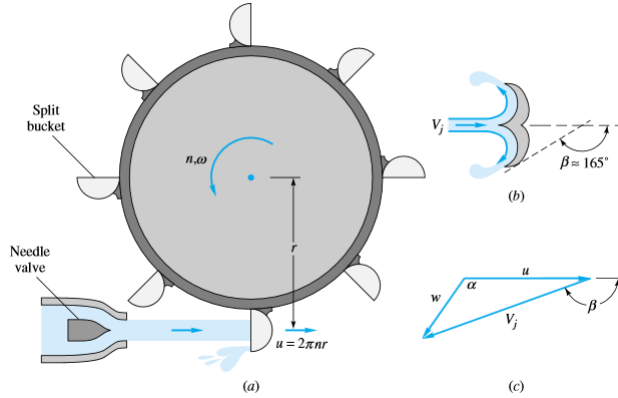


Figure 3.1.1: Pelton turbine [51]: (a) side view of wheel and jet; (b) top view of bucket; (c) typical velocity diagram

A bucket with angle $\beta = 180^\circ$ gives maximum power, but is physically impractical. In practice, $\beta \approx 165^\circ$, or $1 - \cos\beta \approx 1.966$ or only 2 percent less than maximum power.

The theoretical power of a Pelton turbine is maximum when $dP/du = 0$ [51], or when:

$$u = 2\pi nr = \frac{1}{2}V_j \quad (3.1.5)$$

For a perfect nozzle, the entire available head would be converted to jet velocity $V_j = \sqrt{2gH}$. Since there are 2 - 8 percent nozzle losses [51], a velocity coefficient C_v is used ($V_j = C_v\sqrt{2gH}$). The theoretical impulse turbine efficiency then becomes:

$$\begin{aligned} \eta &= 2(1 - \cos\beta)\phi(C_v - \phi) \\ \phi &= \frac{u}{\sqrt{2gH}} = \text{peripheral velocity factor} \end{aligned} \quad (3.1.6)$$

Maximum efficiency occurs at $\phi = \frac{1}{2}C_v \approx 0.47$ [51].

3.1.6 Optimal Rotational Speed

If the absolute velocity at the outlet is zero, the water has transferred all its energy to the runner. Insert the expression for angular velocity from equation 3.1.5 into the equation 3.1.6, with $\cos\beta = 180$, $C_v = 1$ and $\phi = 0.5$ to obtain the optimal angular velocity:

$$\begin{aligned} \eta &= 2(1 - (-1))\frac{u}{\sqrt{2gH}}(1 - 0.5) \\ &= 2\frac{u}{\sqrt{2gH}} = 2\frac{\omega r}{\sqrt{2gH}} \end{aligned} \quad (3.1.7)$$

The optimal angular velocity ω as a function of H , η and the runner diameter D_r :

$$\omega = \frac{\eta\sqrt{2gH}}{D_r} [\text{rad/s}] \quad (3.1.8)$$

and the revolutions per minute is:

$$n = \frac{30\eta\sqrt{2gH}}{\pi D_r} [\text{rpm}] \quad (3.1.9)$$

The optimal angular velocity as a function of the inlet velocity is:

$$\omega = \frac{\eta V_j}{D_r} \quad (3.1.10)$$

3.1.7 Reaction Force on Pelton Turbines

The relative reaction force is the force acting from the jet on a bucket when the runner is rotating. This force can be found by the help of Newton's 2. law [7]:

$$F_R = \rho Q_R (w_1 - w_2) \quad (3.1.11)$$

The relative flow rate is given by:

$$Q_R = A \cdot w_1 = A \cdot (V_j - u) = \frac{\pi \cdot d_j^2}{4} (V_j - u) \quad (3.1.12)$$

Assuming no losses, the following relationship applies for the relative velocities:

$$w_1 = V_j - u \quad w_2 = (V_j - u) \cos \beta \approx -w_1 \quad (3.1.13)$$

The relative reaction force than becomes:

$$\begin{aligned} F_R &= \rho Q_R [(V_j - u) + (V_j - u)] \\ &= 2\rho Q_R (V_j - u) \end{aligned} \quad (3.1.14)$$

The absolute reaction force acting on the Pelton runner is given by the absolute volume flow:

$$F_A = 2\rho Q_A (V_j - u) \quad (3.1.15)$$

When the speed of the runner is zero ($u = 0$), the relative reaction force is equal to the absolute reaction force. The relative reaction force on a bucket F_R is independent of the number of buckets, given that the cross-section of the jet when it is vertical on the bucket is not intersected.

The torque on the runner is:

$$T = \frac{D_r}{2} F_A = \rho Q D_r (V_j - u) \quad (3.1.16)$$

3.2 Kinematics

There are two ways of describing a fluid motion, with either a *Lagrangian* or an *Eulerian* method. In the Lagrangian description, one essentially follows the history of individual fluid particles. For a fluid particle with the position vector \mathbf{x}_0 at some reference time $t = 0$. The particle position is written as $\mathbf{x}(\mathbf{x}_0, t)$, which represents the location at t for a particle whose position was \mathbf{x}_0 at $t = 0$. In the Eulerian description, one follows the history of spatial point \mathbf{x} .

The spacial coordinates \mathbf{x} of a particle may be referred to as its position or place. The coordinates of a particle which is at any position \mathbf{x} at time t :

$$\zeta = \zeta(\mathbf{x}, \mathbf{t}) \tag{3.2.17}$$

are continuous and single valued. The single valuedness of the equation means that a particle cannot be split up and occupy two places nor can two distinct particles occupy the same place.

These are two methods that are widely used to observe and analyze fluid flows, either by observing the trajectories of specific fluid parcels, or by observing the fluid velocity at fixed positions [3].

3.2.1 Particle Paths

The coordinates of the particle ζ , is at any position \mathbf{x} at time t :

$$\zeta = \zeta(x, t) \tag{3.2.18}$$

The equation 3.2.18 is the parametric equation of a curve in space with t as parameter, these curves are called the *particle paths* [3]. Any property of the fluid may be followed along the particle path. The density neighborhood of a particle, can be given as a function $\rho(\zeta, t)$, which is the density as the particle experience. The path line is the trajectory of a fluid particle of fixed identity over a period of time.

At time t the streakline through a fixed point y is a curve going from y to $\mathbf{x}(y, t)$, the position reached by the particle which was at y at time $t = 0$. A particle is on the streakline if it passed the fixed point y at some time between 0 and t . Streaklines are thus the place of points of all the fluid particles that have passed continuously through a particular spatial point in the past [3].

According to Perrig [46] one of the major advantages with meshless Lagrangian particle-based methods once correctly validated, is that they could be useful to obtain the correct streaklines of the various particles. “The analysis of the momentum conservation along these streaklines will provide a complete energy transfer survey in the bucket.”

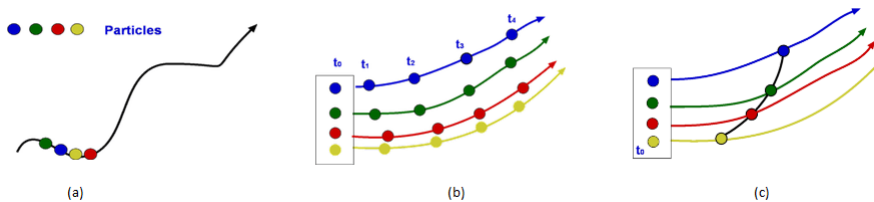


Figure 3.2.2: (a) Timeline is generated by drawing a line through adjacent particles in flow at any instant of time. (b) Pathline is the line traced by a given particle. (c) Streakline concentrates on fluid particles that have gone through a fixed station or point. At some instant of time the position of all these particles are marked and a line is drawn through them [4].

3.3 Computational Fluid Dynamics

Computational fluid dynamics (CFD) is the use of numerical methods and algorithms to solve and analyse fluid mechanics problems. It is, in short, the computation of the interaction of liquids and gases with surfaces defined by boundary conditions.

Ongoing research yields software implementations that improves the accuracy and speed of complex simulation scenarios.

3.3.1 Turbulent Flow

The community of fluid mechanics today has some of the deepest mysteries of contemporary physics, which is turbulent flow. In the late nineteenth century with works conducted by i.a. Reynolds and Boussinesq revolutionized the work of fluid turbulence. Turbulence plays a prevailing part in almost all industrial flows.

3.3.2 Validation and Verification

Developing software necessarily implies a validation step, in order to test the conceptual foundations of the adopted numerical method, as well as to check both implementation and programming for quality.

Validation deals with the assessment of the comparison between sufficiently accurate computational results and the experimental data. The fundamental strategy of validation involves identification and quantification of the error and uncertainty in the conceptual and computational models, quantification of the numerical error in the computational solution, estimation of the experimental uncertainty, and finally, comparison between the computational results and the experimental data.

That is, accuracy is measured in relation to experimental data, our best measure of reality [45].

Oberkampf and Trucano (2002) [49] stated that quantitative assessment of the physical modelling uncertainty requires comparison of the CFD results with high-quality experimental results. And that meaningful validation is only possible with good quantitative estimates of all numerical errors, input uncertainty and uncertainty of the experimental data used in the comparison.

Verification is the process of determining that a model implementation accurately represents the developers conceptual description of the model and the solution to the model. The emphasis in verification is the identification and quantification of insufficient spatial discretization convergence, insufficient temporal discretization convergence, insufficient convergence of an iterative procedure, computer round-off, and computer programming errors, as well as the demonstration of the stability, consistency, and robustness of the numerical scheme [45].

3.4 Meshfree Particle Methods

The meshfree particle methods is developed to provide accurate and stable numerical solutions for integral equations or partial differential equation, with all kinds of possible boundary conditions and with a set of arbitrarily distributed nodes (or particles) without using any mesh that provides the connectivity of these nodes or particles. The meshfree particle methods are intended to remedy problems which the conventional FDM and FEM have, such as problems with free surface, deformable boundary, moving interface (for FDM), large deformation (for FEM), complex mesh generation, mesh adaptively and multi-scale resolution (for both FDM and FEM).

Meshfree particle methods (MPM) treat the system as a set of particles, which represents a physical object or a parcel of the domain. For CFD problems, variables such as mass, momentum, energy, position, etc. are computed at each particle.

There are basically two types of meshfree particle methods: methods based on strong form formulations and methods based on weak form formulations. The strong form method has attractive advantages of being simple to implement, computationally efficient and truly meshfree, because no integration is required in establishing the discrete system equations. However, they are often unstable and less accurate. The weak form method has the advantages of excellent stability, accuracy. However, the weak form method is said not to be truly meshfree, as a background mesh (local or global) is required for the integration of the weak forms [32].

Methods that combines the two grid systems, so called two-grid systems(Lagrangian and Eulerian), exchanges information either by mapping or by special interface treatment between these two types of grids. These approaches work well for many

problems, but are rather complicated and may also cause some inaccuracy in the numerical treatment [32].

3.5 SPH Method

Problems in *computational fluid dynamics* (CFD) are generally solved by employing the conventional grid-based numerical methods such as the *finite difference method* (FDM), *finite volume method* (FVM) and *finite element method* (FEM). These conventional numerical methods have dominated the subject of computational fluid dynamics. An important feature of these methods is the corresponding Eulerian (FDM and FVM) or Lagrangian (FEM) grid that is required as a computational frame to provide spatial discretization for the governing equations. The Eulerian methods are inefficient in treating moving mesh, deformable boundaries, free surfaces, etc [32].

Smoothed Particle Hydrodynamics (SPH) is an adaptive Lagrangian meshfree particle method, first created by Lucy [34] and Monaghan and Gingold [19] for use in astrophysics. There are several benefits with the SPH method over traditional grid-based techniques. SPH conserves the mass without extra computation since the particles themselves represent mass. The method also computes pressure from weighted contributions of neighbouring particles rather than by solving linear systems of equations.

The SPH numerical method is today the most popular meshfree method in scientific literature [50]. SPH is being increasingly used to model fluid motion. It has repeatedly been improved, first to weakly compressible hydrodynamics, then to strongly compressible and truly incompressible [50]. It is based on the use of a "kernel" function which is suitable for representing the derivatives of continuous fields in a discrete form.

3.5.1 The Basics of SPH

The SPH method was developed for hydrodynamics problems that are basically in the form of *partial differential equations* (PDE). The SPH method are based on these key ideas:

- The problem domain is represented by a set of arbitrarily distributed particles.
- The integral representation method is used for field kernel approximation.
- The kernel approximation is further approximated using particles.
- The particle approximation is performed at every time step.

- The particle approximation is performed on all terms related to field functions in the PDEs to produce a set of ODEs in discretized form with respect to time only.
- The ODEs are solved using an explicit integration algorithm.

3.5.2 Smoothing Length

The smoothing length h determines the size of the support domain, thus the the number of particles used to approximate the solution. The size of h directly influences the accuracy of the solution. A small value of h will mean that the number of particles in the support domain is too small to make an accurate SPH approximation while a large h may result in local properties being smoothed out.

3.5.3 Support Domain and Influence Domain

By definition, the support domain for a field point at x is the domain where the information for all the points inside this domain is used to determine the information at the point at x . The influence domain is defined as a domain where a node exerts its influences [32].

For the SPH method, the concepts of support and influence domains for a particle are closely related to the smoothing length h of that particle. The smoothing length h multiplied by a factor determines the support domain or influence domain, where the smoothing function applies.

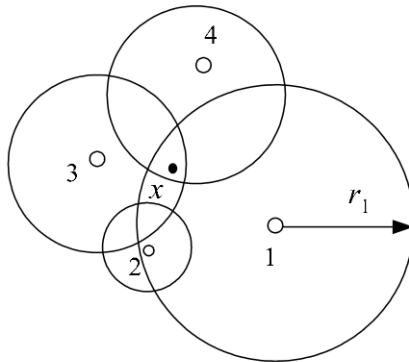


Figure 3.5.3: The concept of support domain. The circle represents the support domain of the corresponding particles [32].

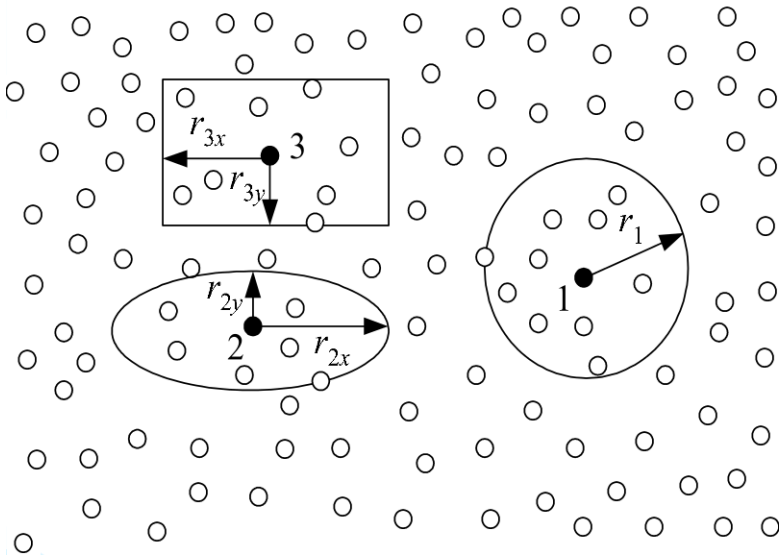


Figure 3.5.4: The concept of influence domain. The solid lines represent the influence domain of the corresponding particles [32].

3.5.4 Integral Representation

The concept of integral representation of a function $f(x)$ is used in the SPH method. With the following identity:

$$f(x) = \int_{\Omega} f(x')\delta(x - x')dx' \quad (3.5.19)$$

where f is a function of properties, of the three-dimensional position vector x , and $\delta(x - x')$ is the Dirac delta function. Ω is the volume of the integral that contains x . In SPH the Delta function $\delta(x - x')$ is replaced by a smoothing function $W(x - x', h)$. W is in this report referred to the kernel function, in SPH literatures it is also known as smoothing kernel function, or smoothing function, or smoothing kernel, or kernel function or simply kernel. In the smoothing function, h , is the smoothing length that define the influence area of the smoothing function W .

3.5.5 Governing equation

There are the three basic governing equations of fluid dynamics which will be presented here, the laws of discrete conservation in the SPH framework

The governing equations for dynamic fluid flow may be expressed by a set of differential equation in Lagrangian form. The set of partial differential equations is the Navier-Stoke equation.

3.5.5.1 Conservation of Mass

The discrete continuity equation:

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^N m_j \mathbf{u}_{ij} \cdot \nabla_i W_{ij} \quad (3.5.20)$$

where N is the number of particles in the support domain of particle i , and m_j is the mass associated with particle j . W_{ij} is the smoothing function of particle i evaluated at particle j , and is closely related to the smoothing length h . The final approximation is written in vector notation with $\mathbf{u}_{ij} = (u_i^\beta - u_j^\beta)$.

The conservation equation is a *particle approximation of density*, first presented by Monaghan [42]. When using this approximation the density of all particles defined initially and the change in density is closely related to the neighbouring particles in the support domain. This is done to derive a particle approximation of the continuity equation. u_{ij} is the relative velocity of a particle pair in the support domain.

There are two ways to conserve mass in SPH [32]; the continuity density approach as in 3.5.20 and the summation density approach:

$$\rho_i = \sum_{j=1}^J m_j W_{ij} \quad (3.5.21)$$

3.5.5.2 Conservation of Momentum

Each form of the equation of motion is consistent with a given form of the of the continuity equation.

$$\frac{Dv_i^\alpha}{Dt} = \sum_{j=1}^J m_j \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_j^{\alpha\beta}}{\rho_j^2} \right) \nabla_i W_{ij} + g^\alpha \quad (3.5.22)$$

$$\frac{Dv_i}{Dt} = \sum_{j=1}^J m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \prod_{ij} \right) \nabla_i W_{ij} + \mathbf{g} \quad (3.5.23)$$

The momentum equation is derived for a Newtonian fluid and the conservation of momentum is assured by using Newton's second law. This version of the momentum

equation is most commonly used in SPH [32]. The viscosity term is solved either by using an artificial viscosity or a turbulent model, as described in section 4.8.1 and 4.8.2 respectively.

There are several notations used for the governing equations in SPH literature. Vila (1999) suggest writing the momentum equations or the equation of motion as [50]:

$$\frac{d\mathbf{u}_a}{dt} = -\frac{1}{\rho_a} \sum_b V_b (p_a \mathbf{R}_a + p_b \mathbf{R}_b) w'_{ab} \mathbf{e}_{ab} + \mathbf{g} \quad (3.5.24)$$

where viscous forces are omitted, $\mathbf{e}_{ab} = \frac{\mathbf{r}_{ab}}{r_{ab}}$ is the unit vector being directed from particle a to particle b and the vector \mathbf{R} is referred to as the systems centre of mass.

3.5.5.3 Conservation of Energy

Since the mass of each particle is constant, the derivative of the total internal energy is written:

$$\frac{DE_{int}}{Dt} = \sum_j m_j \frac{De_{int,a}}{Dt} \quad (3.5.25)$$

$$\frac{De_i}{Dt} = \frac{1}{2} \sum_{j=1}^J m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \prod_{ij} \right) u_{ij} \nabla_i W_{ij} \quad (3.5.26)$$

The conservation of energy is based on the first law of thermodynamics where the change in internal energy is equal to the heat added to the system minus the conducted work. As there is no heat added to the system in the SPH formulation the only source of change of internal energy in the fluid cell consists of the work done by the body forces. The work done by the body forces consist of the isotropic pressure multiplying the volumetric strain and the energy dissipation due to the viscous shear forces. Where $\sigma^{\alpha\beta}$ is the total stress tensor from equation 3.5.22.

Several alternative conservation equations exist, but there are no noticeable differences between them [32].

3.5.6 Kernel Functions

The use of different kernels is the SPH analogue of the use of different difference schemes in finite difference method. In the SPH literatures, the kernel approximation is often said to have h^2 accuracy or second order accuracy [42]. A kernel

function should satisfy a number of conditions. One of these is the Delta function property, as mentioned in 3.5.4 [32].

3.5.7 Particle Approximation and Interpolation

In the SPH method, the entire system is represented by a finite number of particles that carry individual mass and occupy individual space. The continuous integral representation concerning the SPH kernel approximation can be converted to discretized forms of summation over all the particles in the support domain. The corresponding discretized process of summation over particles is commonly known as particle approximation in the SPH literatures [32].

From equation 3.5.19, we get a scalar field A:

$$A(x, t) = \int_{\Omega} A(x', t) W_{ij} dx' \tag{3.5.27}$$

which defines a discrete interpolation of the field. The discrete sum covers all the particles. Each particle interacts with a finite number of adjacent particles. This elements are significantly higher than the number of the elements adjacent to an element in the frame of the mesh methods, which is a weakness in the SPH method.

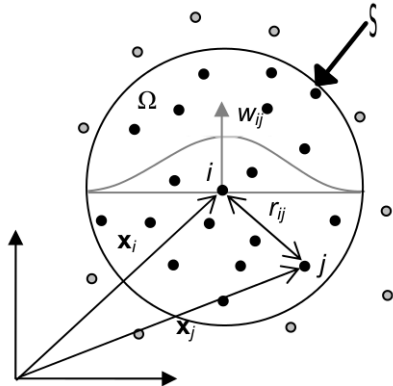


Figure 3.5.5: Particle approximations using particles within the support domain[S] of the smoothing function[$W(x_i - x_j, h)$] for particle i. The support domain is circular with a radius of κh [32].

The use of particle summations to approximate the integral, is a key approximation that makes SPH method simple without using a background mesh for numerical integration. The particle approximation is one of the major reasons for the SPH

method being particular popular for dynamic fluid flow problems. Since the particle approximation introduces the mass and density of the particle into the equation.

3.5.8 Incompressibility

When solving incompressible flow problems in SPH two different approaches are usually adopted, a weakly compressible fluid concept or a projection method.

Weakly compressible SPH (WCSPH), a compressible flow solver is used along with an artificial speed of sound which is usually at least 10 times larger than the maximum velocity calculated in the flow field. This high value for the speed of sound guaranties that the variation in density remains less than one percent.

Although the WCSPH is a simple and practical approach, it suffers from spurious oscillations in the pressure and density fields [17]. When the Reynolds number is relatively high, depending on the problem, the pressure gradient may become the dominant force and non-physical pressure distribution can interfere with the solution.

An equation of state is required to describe the variation of density with pressure in the WCSPH. A simple and frequently used equation of state is [17].

$$P - P_0 = c^2(\rho - \rho_0) \quad (3.5.28)$$

in which $c = (\partial P / \partial \rho)_s^{1/2}$ is the artificial speed of sound defined for an isentropic process. To satisfy this condition, it is necessary to have a large speed of sound. Often, a speed of sound of 10 times greater than the maximum velocity of the fluid is sufficient [17].

In the second approach, the projection method, the pressure field is obtained by solving a Poisson equation. Recently studies [17], which have compared these two approaches on various incompressible flow test cases, have concluded that WCSPH produces unreliable pressure fields especially on coarse resolutions.

One can substitute the density in the left-hand side of the mass conservation equation 3.5.20 by its value from the equation of state 3.5.28. The pressure equation is transformed into the Poisson equation.

3.5.9 Boundary Treatment

When applying the SPH method to real flow modelling implies usually the presence of solid walls. When a fluid particle comes close to a wall, a void space gradually arise within its range of influence and this leads to a pressure force. The pressure force is insufficient to force the fluid particle to remain within the fluid domain: thus the fluid particles penetrate the wall.

When SPH was developed over 30 years ago, the technique was designed for astrophysics simulating galaxy formation, and in space there are no boundaries. But in CFD all boundaries are either open or closed (solid wall). The figure 3.5.6 illustrates the problem, there are no fluid particles in the boundary. Therefore artificial particles must be modelled.

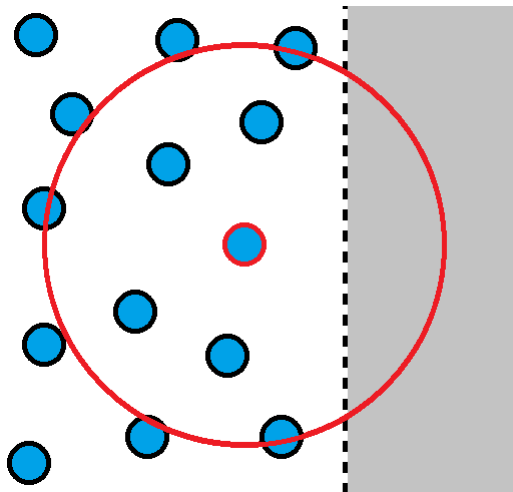


Figure 3.5.6: Interior fluid particles at the bound.

Boundary conditions are one of the major weaknesses for practical use of SPH in real cases [37]. There exist three major methods used in SPH to model boundary particles, the solid wall particles, the boundary repulsive force and the mirror particle or the ghost particle.

3.5.10 Key Issues in SPH

3.5.10.1 Stability

The integral representation has a smoothing effect that behaves as a weak form formulation. Thus, the equations are no longer required to hold absolutely and has instead weak solutions. This is equivalent to formulating the problem to require a solution in the sense of a distribution. The weak form formulation are usually very stable.

Numerical instabilities can occur in situations where a large quantity of compressible fluid is constrained, and the pressure can become extremely high on a small area [27]. The high pressure can then create turbulence reactions in the water.

3.5.10.2 Consistency

For an SPH approximation to converge the solution must approach the exact solution when the nodal distance approach zero. To ensure the convergence, the SPH kernel function must satisfy a certain degree of consistency. In solving any partial differential equation based on a weak form formulation, there is a minimum consistency requirement for ensuring the convergence of the discretized equation system.

The kernel consistency is related to the SPH kernel approximation. Even if some kernel consistency condition is satisfied do not this mean that the corresponding SPH particle approximations in discretized form also satisfy the discretized consistency condition. This discrepancy between the particle approximation and the kernel approximation is termed as particle inconsistency [33].

The discretized consistency condition are not always satisfied. An example of this is when the particles are at or near the boundary of the problem domain, so that the support domain intersects with the boundary. Another case is observed when the particles are irregularly distributed [32].

Inconsistency causes errors in the SPH solution, as a small error in the density results in a large fluctuation in pressure during simulation. The system then becomes unstable. In order to solve this problem, a density diffusion term in the equation for the mass conservation is introduced [48].

3.5.10.3 Smoothing Length

The smoothing length, h , determines the size of the support domain and therefore also the number of particles that is used in the approximation of the kernel function. Hence, the smoothing length is influencing the accuracy of the solution and the efficiency of computation. The smoothing length, h , is important when choosing the right time step, 3.5.10.4, dt , for the time integration. The most common use of smoothing length is to update the smoothing length according to the average density [32]:

$$h = h_0 \left(\frac{\rho_0}{\rho} \right)^{1/d} \quad (3.5.29)$$

where h_0 and ρ_0 are the initial smoothing length and the initial density. d is the number of dimensions.

3.5.10.4 Time Step

A good the time step has to be determined in order to ensure a stable time integration. Standard time-step control can involve the Courant condition, the force term

and the viscous diffusion term [42]. When water is modelled as weakly compressible fluid a very small time step are required, according to the Courant condition.

3.5.10.5 Kernel Functions

One of the central issues for the meshfree methods is how to effectively perform function approximation based on a set of nodes scattered in an arbitrary manner without using a predefined mesh or grid that provides the connectivity of the nodes.

The SPH method employs the integral representation using a kernel function. The kernel function is important since it not only determines the pattern for the function approximation, but also defines the dimension of the support domain of particles. Thus determines the consistency and hence the accuracy of both the kernel and particle approximations.

3.5.10.6 Boundary Treatment

SPH has been hampered by the problem of particle deficiency near or on the boundary. For particles near or on the boundary, only particles inside the boundary contribute to the summation of the particle interaction, and no contribution comes from outside since there are no particles beyond the boundary [32]. The boundary conditions are probably the worst culprit of all problems for free-surface flow.

3.5.10.7 Particle Penetration

Penetration occurs in SPH because the method does not require that the velocity field must be single valued. Hence, two or more particles, with different velocities, may occupy the same position. For low Mach number flows with smooth velocity fields this is seldom a problem, but at high Mach number flows the problem is severe [41].

Chapter 4

DualSPHysics

The DualSPHysics is a parallel version of the FORTRAN code implemented in C++ and CUDA language to carry out simulations on both the CPU and GPU.

SPHysics and DualSPHysics are open-source SPH models developed by researchers at the Johns Hopkins Univeristy, the University of Vigo, the University of Manchester and the University of Rome.

The SPHysics Fortran code(previous non-parallel version) has been validated for different problems [14] [21].

The new object-oriented program provides better overview, easier maintenance and modification, lower runtime and a good error control [14].

4.1 Kernel Functions

The SPH metod uses kernel functions, as described in 3.5.6. Thus, the physical quantity of any particle can be obtained by summing the relevant properties of all the particles which lie within the range of the kernel.

Available functions in DualSPHysics [14]:

- Cubic Spline kernel [Monaghan and Lattanzio, 1985]
- Quintic Wendland kernel [Wendland, 1995]

Cubic Spline kernel

$$W(r, h) = \alpha_D \begin{cases} 1 - \frac{3}{2}q^2 - \frac{3}{4}q^3 & 0 \leq q \leq 1 \\ \frac{1}{4}(2 - q)^3 & 1 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases} \quad (4.1.1)$$

where α_D is $10/(7\pi h^2)$ in 2D and $1/(\pi h^3)$ in 3D.

Quintic Wendland kernel

$$W(r, h) = \alpha_D \left(1 - \frac{q}{2}\right)^4 (2q + 1) \quad 0 \leq q \leq 2 \quad (4.1.2)$$

where α_D is $7/(4\pi h^2)$ in 2D and $7/(8\pi h^3)$ in 3D.

Results show that the best compromise between accuracy and time computation cost is reached by the use of the Wendland kernel. In general, the higher the order of the kernels, the greater the accuracy of the SPH scheme [12].

Kernels depend on the smoothing length, h , and the non-dimensional distance between particles given by $q = r/h$, r being the distance between particles i and j [12].

4.1.1 Kernel Gradient Correction

To ensure consistency and normalization DualSPHysics have implemented kernel gradient correction. In order to ensure that the gradient of a velocity field is correctly evaluated, the gradient correction proposed by (Bonet and Lok, 1999) is implemented [14].

4.2 Smoothing Length

The smoothing length determines the size of the support domain. In DualSPHysics the support domain is defined as a sphere.

In analytic geometry, a sphere with centre (x_0, y_0, z_0) and radius r is the locus of all points (x, y, z) such that:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \quad (4.2.3)$$

In DualSPHysics:

$$h = \kappa \cdot \sqrt{dx^2 + dy^2 + dz^2} \quad (4.2.4)$$

where $\kappa = 0.866$ is the default value in DualSPHysics, the smoothing length h is constant for all particles and $dp = dx = dy = dz$. The radius for the support domain (sphere) is h/κ .

From section 3.5.2, the smoothing length has an impact on the quality of the solution. In 3D the optimal h is very hard to predict. The smoothing length produces

two contradictory effects. The higher h/dp , the larger the number of particles involved in the discrete sum, which tends to reduce the statistical inaccuracy of the discrete estimation. On the other hand, the higher h , the broader the integration domain, which disrupts the estimation of the interpolated quantity at the point of interest and causes all the greater errors. From these two conditions, h should tend toward zero whereas h/dp should tend toward infinity. The particle size shall then obviously be as small as possible. The optimal value of the h/dp ratio, is not universal [50].

In DualSPHysics the h/dp ratio is constant.

$$\begin{aligned}\frac{h}{\kappa} &= \sqrt{dp^2 + dp^2 + dp^2} \\ &= \sqrt{3dp^2} \\ \frac{h}{dp} &= \sqrt{3} \cdot \kappa\end{aligned}\tag{4.2.5}$$

4.3 Time stepping

Two numerical schemes are implemented in DualSPHysics, the Verlet algorithm and the Symplectic algorithm [14].

The most used one is the Symplectic time integration algorithm, which is time reversible in the absence of friction or viscous effect, and hence represent a very attractive option for meshfree particle schemes.

The time step in DualSPHysics is controlled by the CFL condition, the forcing terms and the viscous diffusion condition. The time step is dynamic and calculated according to the solution presented by Monaghan and Kos [43].

A variable time step Δt is calculated according to:

$$\Delta t = a \cdot \min(\Delta t_f, \Delta t_{cv})\tag{4.3.6}$$

where Δt_f is the time step based on force, Δt_{cv} is the time step based on velocity and a is a constant to calculate the time step (in DualSPHysics it can vary between 0.1 - 0.5). The CFL condition is heavily based on the particle spacing.

4.4 Particle Movement

Particles are moved in DualSPHysics using the XSPH variant. XSPH was introduced by Monaghan, where "X" is the unknown factor, i.e. the additional factor

added to the velocity [42]. The XSPH variant moves a particle with a velocity that is closer to the average velocity in its neighborhood:

$$\frac{dr_a}{dt} = \hat{v}_a = v_a + \varepsilon \sum_b m_b \left(\frac{v_{ba}}{\bar{\rho}_{ab}} \right) W_{ab} \quad (4.4.7)$$

with $\bar{\rho}_{ab} = (\rho_a + \rho_b)/2$ and $0 \leq \varepsilon \leq 1$ as a constant. XSPH is turn off if $\varepsilon = 0$. In DualSPHysics the default value of ε is 0.5.

No dissipation is introduced by XSPH, but there is increased dispersion. The XSPH prevents particles with different velocities from occupying the same location and keeps fluids orderly in high speed flows. This has been proven in simulations of nearly incompressible fluids such as water [42].

4.5 Boundary Conditions

In DualSPHysics the boundary is described as a set of discrete boundary particles.

DualSPHysics uses dynamic boundary conditions(DBS), where the boundary particles are forced to satisfy the same equations as fluid particles. The boundary particles are fixed in position or move according to a external function. When fluid particles approach the boundary, the density of the boundary particles increases and this results in a pressure increase.

This boundary treatment present some problems, due to the repulsion mechanism. In order to reduce overestimated repulsion, Hughes and Graham suggested to not update the pressure values of the boundary particles at each time step. Thus, the system evolves for several time step while the repulsion force is kept constant [20]. In DualSPHysics it is optional to choose interval of steps between update of the density of the boundaries. The dynamic boundary condition implemented in DualSPHysics is validated by an oversimplified geometry where a single particle impinges a boundary.

4.6 Particle Interaction

The linked list algorithm is used when searching for the nearest neighbouring particles to a particle. A linked list is constructed using pointers in C++. The Cubic Spline kernel have a mesh spacing equal to $2h$, and work well with linked list. Because particle only interacts with particles in its own and the eight neighbouring(2D) cells. Since four of the neighbouring cells already have been resolved, it is only necessary to search in the remaining four cells. This reduces the computation time [12].

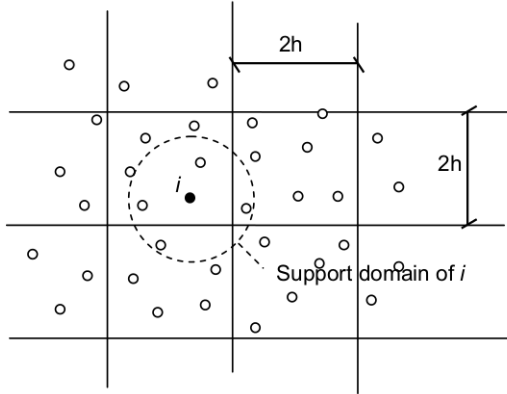


Figure 4.6.1: The circle illustrates the number of particles in the linked list with the Cubic Spline kernel.

In DualSPHysics there exist different cell division of the domain. The domain is split into cells of size $(2h * 2h * 2h)$ to reduce the neighbour search to only the adjacent cells or the domain can be split into cells of size $(h * h * h)$.

4.7 Weakly Compressible SPH

In weakly compressible SPH, the pressure is related to the density by a quasi-incompressible equation of state.

For incompressible flows in the standard SPH method, the equation of state of the fluid will lead to prohibitive time steps that are extremely small [32]. Thus, the pressure term in the momentum equation is a major task.

In fluid mechanics, the Tait equation is an equation of state, used to relate liquid density to pressure. In 1999 Monaghan and Kos [43] introduced the artificial compressibility to produce the time derivative of pressure from Tait's equation. The relationship between pressure and density:

$$P = B \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right) \quad (4.7.8)$$

$$B = \frac{c_0^2 \rho_0}{\gamma} \quad (4.7.9)$$

In DualSPHysics:

$$c_0^2 = C_{coef}^2 \cdot g \cdot h_{SWL} \quad (4.7.10)$$

Equation (4.7.9) and (4.7.10) gives:

$$B = \frac{C_{coef}^2 g \rho_0 h_{SWL}}{\gamma} \quad (4.7.11)$$

where,

γ is constant depending on the problem, for water $\gamma = 7$ [43]

ρ_0 is the reference density

c_0 is the speed of sound at the reference density

This technique is very useful combined with the XSPH technique, described in section 4.4. When applied to XSPH, the unphysical penetration can be efficiently reduced between approaching particles.

The speed of sound c_0 is recommend a minimum relation of $10v \approx c_0$ [43], where v is the bulk flow. This is in order to keep density variation within 1%.

An incompressible SPH algorithm is not yet implemented in DualSPHysics, but they exist. It require larger computational effort at each time step, but the speed of sound is no longer the deciding factor.

This use of a compressible fluid is similar to Chorin's artificial compressibility method [43].

4.8 Viscosity

In DualSPHysics there are two techniques to treat the viscosity.

- Artificial viscosity (Monaghan, 1992)
- Laminar viscosity + SPS turbulence model (Dalrymple and Rogers, 2006)

The artificial viscosity approach being the more common approach as it is the oldest.

4.8.1 Artificial Viscosity

The artificial viscosity was developed by von Neumann and Richtmyer(1950), to represent the transformation of kinetic energy into heat energy as a form of viscous dissipation [32]. The quadratic von Neumann-Richtmyer artificial viscosity and the linear artificial viscosity are widely used today in removing numerical oscillations in hydrodynamics simulations [32]. The Monaghan type artificial viscosity Π_{ab} is most widely used so far in the SPH literatures [32].

Monaghan's artificial viscosity [42] [20] is formulated as:

$$\prod_{ab} = -\frac{\alpha\mu_{ab}\bar{c}_{ab}}{\bar{\rho}_{ab}} \quad (4.8.12)$$

The viscosity \prod_{ij} is equal to 0, when $v_{ij} \cdot r_{ij} > 0$. Which is the SPH equivalent of the condition $\nabla \cdot v > 0$ [42].

The viscosity term in the SPH equation of momentum, in section 3.5.23 is describe with \prod_{ij} . Where α is an empirical coefficient between 0.001-0.1. The approach has several advantages as it keeps particles from penetrating and keeps free surface flow numerical stable. Disadvantages with this approach, it is too dissipative and affects the shear in the fluid [15].

4.8.2 Laminar Viscosity and Sub-Particle Scale (SPS) Turbulence

In many cases the artificial viscosity is too dissipative and affects the shear in the fluid. This is particularly important when using SPH to capture coherent turbulent structures. Dalrymple and Rogers(2006) [15] replaced the standard SPH artificial viscous formulation by introducing a sub-particle scaling technique using the Large Eddy Simulation method (LES) approach, similar to that used in incompressible flows.

4.8.3 Shepard Density Filter

Shepard filter is performed to ensure a smooth free surface [14]. The Shepard filter is a quick and simple correction to the density field, and the following procedure is applied every(20-50) time step [22].

$$\rho_i^{new} = \sum_j \rho_j \hat{W}_{ij} \frac{m_j}{\rho_j} = \sum_j m_j \hat{W}_{ij}$$

where

$$\hat{W}_{ij} = \frac{W_{ij}}{\sum_j W_{ij} \frac{m_j}{\rho_j}} \quad (4.8.13)$$

4.9 GPU

The appearance of the graphics processing unit(GPU) has changed the capability of scientific computing, which was hitherto only available with expensive high-performance computing(HPC) facilities. With DualSPHysics using a single GPU,

results showed a speedup rate of 64 (experience has shown with other tests that this speedup can be even higher) [21] [11].

4.10 XML

The extensible markup language(XML) is a textual data format compatible with any hardware and software. The dualSPHysics use an XML file containing all the parameter of the system configuration and its execution such as key variables, time step, etc. A C++ code named GenCase, define the initial configuration of the simulation of the execution in DualSPHysics. All this information is contain in an XML file.

4.11 Post-Processing

The *MeasureTool* code allows different physical quantities at a set of given points to be computed, to compare experimental and numerical values.

The MeasureTool code interpolates the values of the neighbouring particles around a given position using different kernels. The distance of interpolation can be the size of the kernel or can be changed.

Flow visualization in fluid dynamics is used to make the flow patterns visible, in order to get a qualitative or quantitative flow information. When using a large number of particles, the visualization of the simulation can be improved by representing surfaces instead of particles. The visualization code *IsoSurface* uses the marching cubes algorithm to extract a polygonal mesh of an isosurface from a 3-D scalar field.

4.12 Future Improvements in DualSPHysics

The developers of DualSPHysics plans to implement improvements in future editions, which may be useful for simulations in Pelton turbines. [14]:

- **New MPI implementation with load dynamic balancing**
Implementing good load balance in MPI parallel program is very important. It may reduce running time and improve performance of MPI parallel program.
- **MultiGPU**
A multi-GPU program benefits from increased performance.

- **New improvements in GenCase and other post-processing tools**
The new version of GenCase can read position of points from a file. In the case of creating a cylinder with concentric coordinates this can be useful.
- **SPH-ALE with Riemann Solver**
The use of an Riemann solver in the SPH-ALE method allows sharp interfaces between fluids. One of the major advantages of the method is that it remains stable for very high density ratios, without adopting non-physical speed of sound [31]. The arbitrary Lagrangian-Eulerian (ALE) is a finite element formulation. A SPH-ALE implementation by Marongiu [39] showed outstanding results as regards of prediction of the pressure at fixed or moving walls. The main advantage of introducing Riemann solvers into SPH is that the pressure and velocity fluctuations present in so many of the SPH schemes for water are removed [50].
- **Inlet/outlet flow conditions**
In version 2.01 the initial values of total number of particles are constant. Particles are permit to exit the solution domain, but not enter.
- **Multiphase**
A multiphase(air-solid-water) solver is essential for simulating the free surface flows encountered in Pelton turbines.
- **δ -SPH**
A new SPH formulation developed at CNR-INSEAN, in the framework of the NEXTMUSE project, allows to avoid pressure instabilities which typically develop using the standard model.

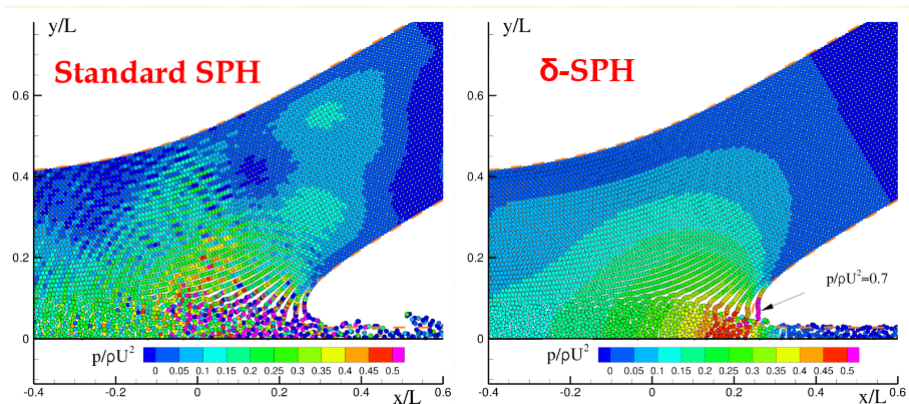


Figure 4.12.2: Case with a non viscous jet impinging on a flat plate, $\theta = 30^\circ$ [10]

- **Modified Virtual Boundary Conditions**
Boundary conditions are one of the greatest challenges associated with SPH today, and intensive research are being called for.

Chapter 5

Water Jet Impinging Normally on a Fixed Plate

The impingement of a water jet on a fixed plate has been considered as a benchmark test for assessment of meshfree particle methods [28] [40]. After the jet impact and release of water-hammer (shock) pressure the flow regime becomes steady and the stagnation pressure at the stagnation point is obtained from the Bernoulli equation.

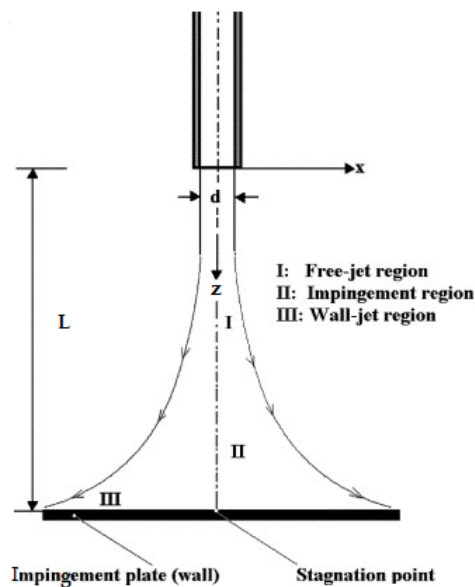


Figure 5.0.1: Impinging Jet Configuration [26]

5.1 Description

A jet impinging an orthogonal plate problem. Let us consider a circular jet of diameter d emerging from a nozzle with a uniform velocity of U_0 located at a distance of L above a smooth flat plate, as illustrated in the figure 5.0.1. The jet from the nozzle impinges normally on the plate and spreads radially outwards as a radial wall jet. Experiments on this type of radial wall jet have been performed a number of times [47].

5.2 Air-Water Interaction

High velocity turbulent water jets discharging into the atmosphere are often used in hydraulic structures to dissipate energy, and are used with Pelton turbines. Considering a water jet discharging into air, the pressure distribution is quasi-uniform across the jet and the buoyancy effect is zero in most cases.

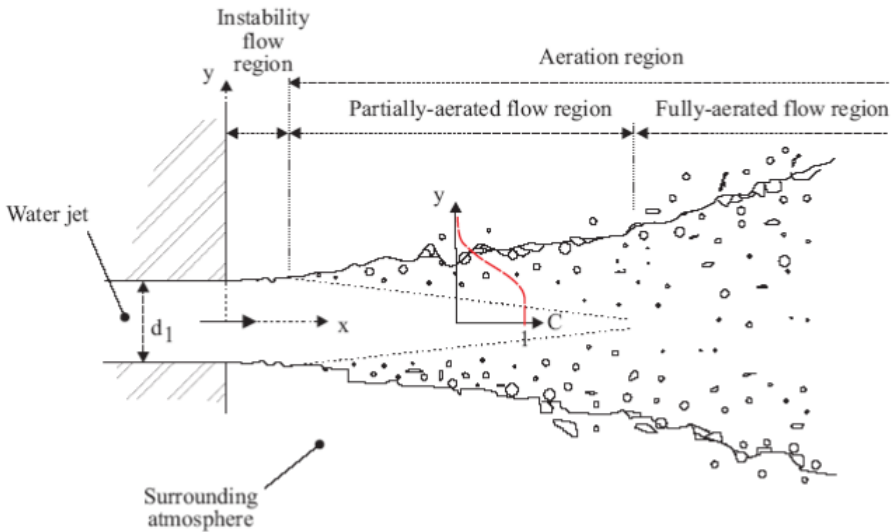


Figure 5.2.2: Definition Sketch [8]

A multiphase flow model is required for accurate simulation of the free-jet region. In DualSPHysics there exist no possibility to model air particles, hence no manner to simulate air-water interaction. The developers of DualSPHysics plan to implement this in future version [14].

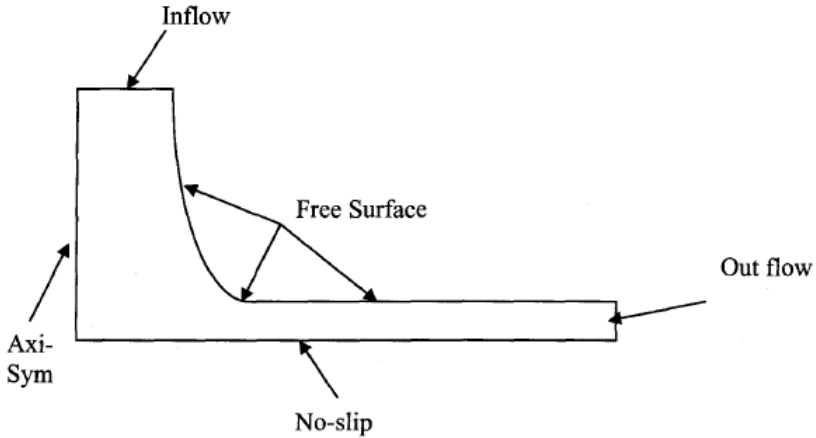


Figure 5.2.3: Jet Impingement Region [36]

5.3 Pre-Processing

In DualSPHysics the pre-processing is done with the C++ code GenCase. All the initial configuration of the simulation is define in a XML file. GenCase implements the initial configuration of the simulation define in the XML, and construct any object using particles. Predefined shapes can easily be added to the simulation, like boxes and cylinders used in this case.

5.3.1 Setup

The study done by Molteni and Colagrossi [40] on a jet impinging on a flat plate was used as a reference, where the speed of sound c_0 was set equal to 12 times the inflow velocity of the jet. Three ratio values between particle space (dp) and jet diameter (d) was used, 10, 20 and 40. The standard artificial viscosity coefficient (α) was equal to 0.1 when used.

In this study, since DualSPHysics cannot simulate air-water interaction a small L was chosen, and other variables are chosen according to previous studies.

Time algorithm	Symplectic
Viscosity formulation	Artificial
Kernel function	Wendland
Reference water density ρ_0	1000 [kg/m ³]
Uniform jet velocity U_0	40 [m/s]
Jet diameter d	0.2 [m]
Initial distance L	0.04 [m]

Table 5.3.1: Setup variables

The water jet is a cylinder filled with water particle and have a diameter (d). The plate is a square box with the size 0.4 x 0.4 x 0.06 [m], and is filled with bound particles.

5.3.2 Run Script in MATLAB

A Matlab script was created for automatically run of a number of simulations with combination of DualSPHysics parameters.

5.4 Parameter study

A parameter study is a systematic way to vary a number of model parameters, and have the system automatically run simulations for each combination of the parameters. In this case a water jet impinging on a flat plate.

The simulation was organized by simulation numbers. For a detailed correlation between parameters and simulation numbers, see appendix D.

5.4.1 Selecting of parameters

The following parameters were selected:

C_{coef}	Particle Spacing(dp)	α	Shepard	DBCSteps	Minimum Time Step
40	0.0015	1/125000	0	1	0.5e-8
100	0.002	1/12500	50	50	1.0e-8
200	0.003	1/1250		100	
400		1/125			

Table 5.4.2: Simulated parameters

A summary of simulation parameters is given in table D.0.1.

5.4.1.1 Speed of Sound

In the weakly compressible model an equation of state is used to determine the fluid pressure, and the compressibility is adjusted to slow the speed of sound to ensure that the time step is reasonable (using a CFL condition based on the speed of sound).

Another limitation on compressibility is to restrict the sound of speed to be at least ten times faster than the maximum fluid velocity, thereby keeping density variations to less than 1 % [20].

In this study the following ratio between the artificial speed of sound and the initial uniform fluid velocity (c_0/U_0) was selected; 2,5,10 and 20. The artificial speed of sound is determined by the speed of sound coefficient (c_{coef}).

5.4.1.2 Repulsion Mechanism

As mentioned in section 4.5 boundary treatment present some problems in Dual-SPHysics, due to repulsion mechanism. Hughes and Graham correction in Dual-SPHysics is a tool to prevent this from happening. This was used in this study together with the Shepard density filter.

5.4.1.3 Viscosity

Viscosity has an almost negligible effect on water flow because the fluid remains nearly incompressible [43].

In the artificial viscosity model α is a free parameter depending on the problem, from equation 4.8.12.

5.5 Validation

Validation is the process of successively testing a software functions in the basis of theoretical solutions or experimental data. In order to determine the degree to which a model is an accurate representation of the real world.

5.5.1 Stagnation pressure

The compressibility of water is a function of pressure and temperature. The low compressibility of liquids, and of water in particular, often leads to water being assumed as incompressible.

The stagnation pressure for incompressible flow can be derived from the Bernoulli Equation:

$$P_{total} = \frac{1}{2}\rho v^2 + P_{static} \quad (5.5.1)$$

5.5.2 Pressure Distribution

Several experimental results exist for a radial wall jet produced by impinging circular jet [47]. The pressure distribution close to the impingement point is very well approximated by analytical stagnation-flow solutions.

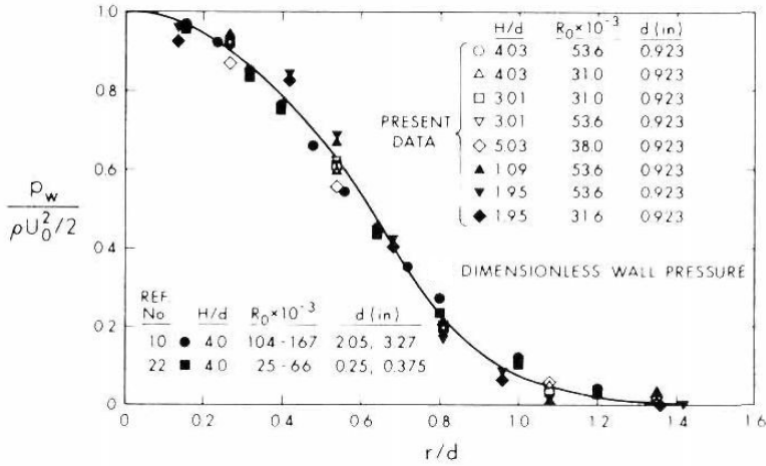


Figure 5.5.4: Radial pressure distribution [6]

5.6 Post-Processing

Computational fluid dynamics (CFD) are nowadays used to calculate and model fluid flow and heat transfer in complex geometries. Due to the increase of calculation capacity of computers more ambitious numerical calculations becomes possible, and the file output of these programs can be enormous. Large amounts of raw data requires post-processing before it can be used. Post-processing involves interpret simulated or observed data, i.e. representation of results.

Post-processing often consists of visualization of data, or representation of data from simulations. Flow visualization is an area of scientific visualization. There are several different techniques to visualize scientific data, with isosurface reconstruction and direct volume rendering being the more common.

5.6.1 MeasureTool

In this case the pressure at the centerline from the stagnation point $(0,0,0.01)$ to the outflow point $(0.2,0,0.01)$ with $\Delta x = 0.002$ was calculated. The obtained output file in .asc (a computer filename extension) contain all the numerical values of the pressure distribution along the plate.

5.6.2 ParaView

ParaView is an open source mutiple-platform application for scientific visualization. It is an application built on top of the Visualization Tool Kit (VTK) libraries.

Here, ParaView has been used to investigate geometries, diffusion and penetration, and for flow visualization. Since MeasureTool uses kernel interpolation mean variables over time were plotted in ParaView for a certain number of particles at the stagnation region.

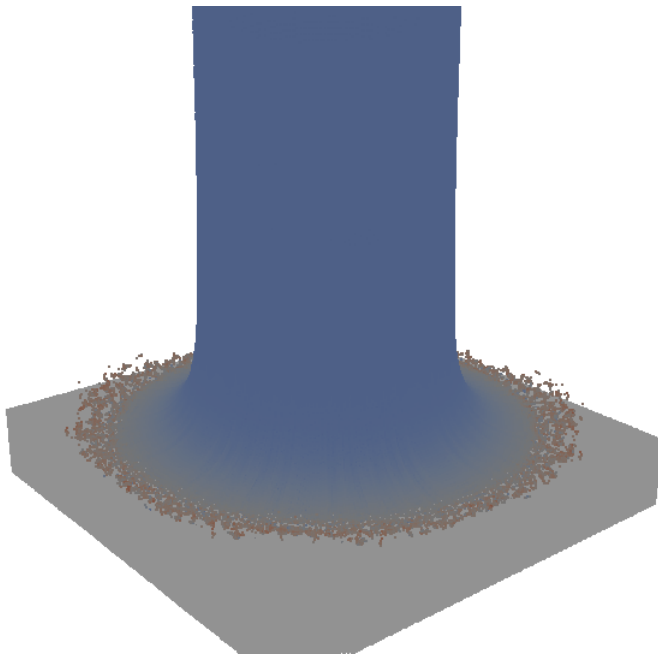


Figure 5.6.5: A water jet impinging a plate plotted in ParaView.

5.6.3 MATLAB

ASCII or .asc files from MeasureTool with all the numerical values of the pressure distribution along the plate are imported to Matlab. In a Matlab script the data

are processed and visualized.

Figure 5.7.15 shows a comparison between the interpolation kernel in MeasureTool and mean values in ParaView.

5.7 Results

The results from almost 300 simulations are presented in this section. The simulations were run on the GPU Nvidia Tesla C2075.

5.7.1 Absolute deviation

For all simulation the absolute deviation was calculated at each write interval ($=0.0001$) while the jet was impinging the plate, and finally a mean deviation over all these time steps was calculated, as showed in 5.7.6.

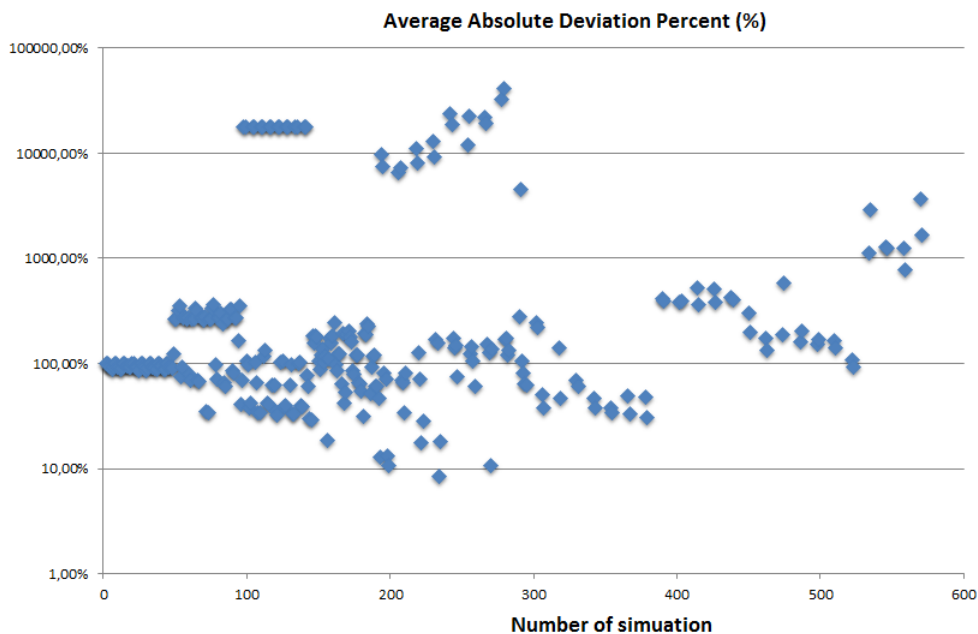


Figure 5.7.6: Absolute deviation over time

A deviation at $\sim 100\%$ indicate that the pressure measurement at the stagnation point is approximately zero (at least at some time steps when the jet is impinging the plate), due to repulsion effects. All simulation with a lower deviation than 90 % are presented in figure 5.7.7.

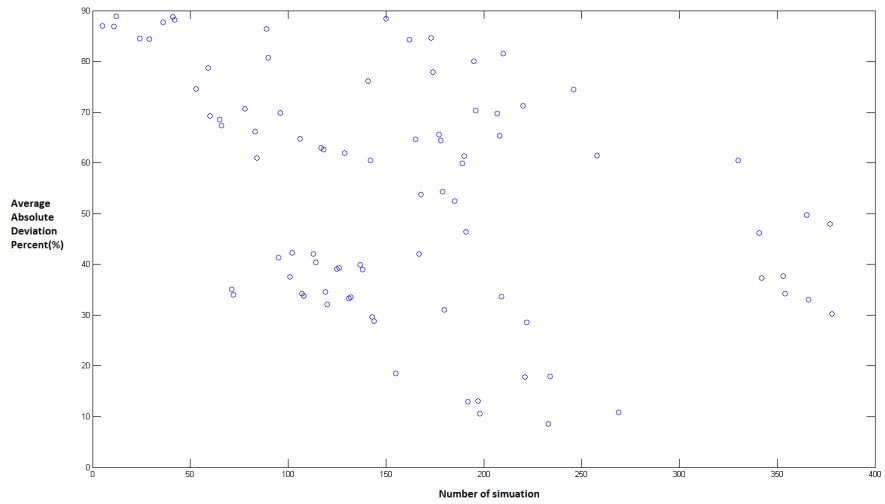


Figure 5.7.7: Absolute deviation over time

5.7.2 Stagnation Point Over Time

Time history of analytical and calculated pressure at the stagnation point is evaluated with different ratio between the artificial speed of sound and the initial uniform fluid velocity.

From these results, a ratio c_0/U_0 between five and ten is in good agreement to

theory, according to literature this ratio should be at least ten [50].

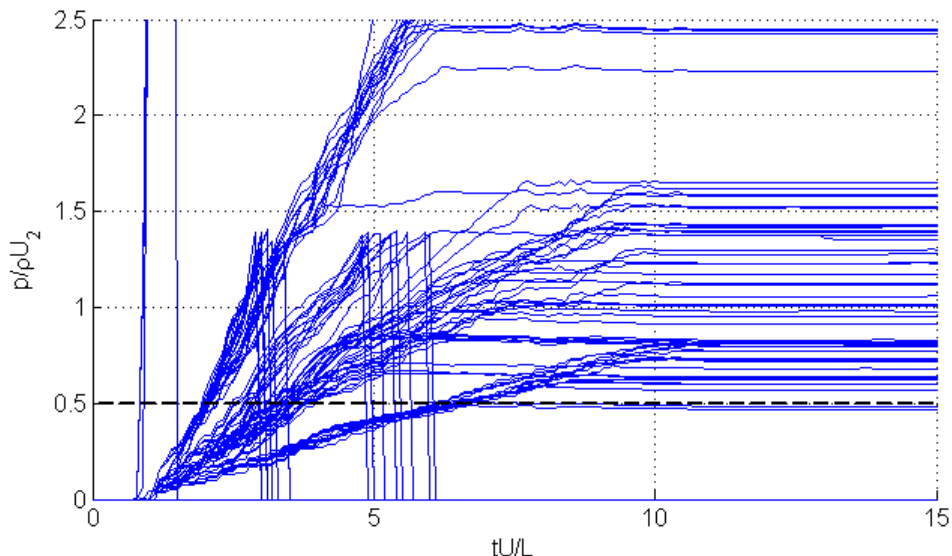


Figure 5.7.8: Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 2$ and the dashed line is the theoretical pressure.

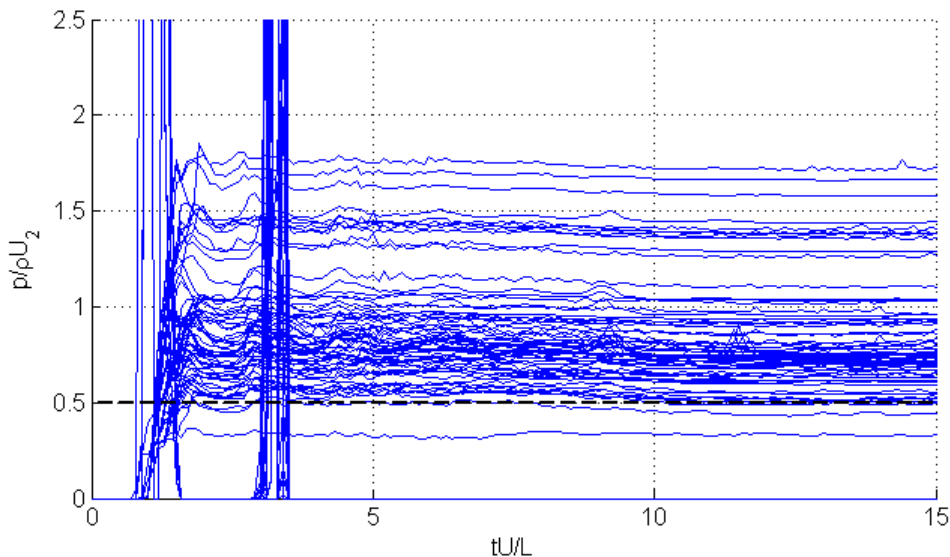


Figure 5.7.9: Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 5$ and the dashed line is the theoretical pressure.

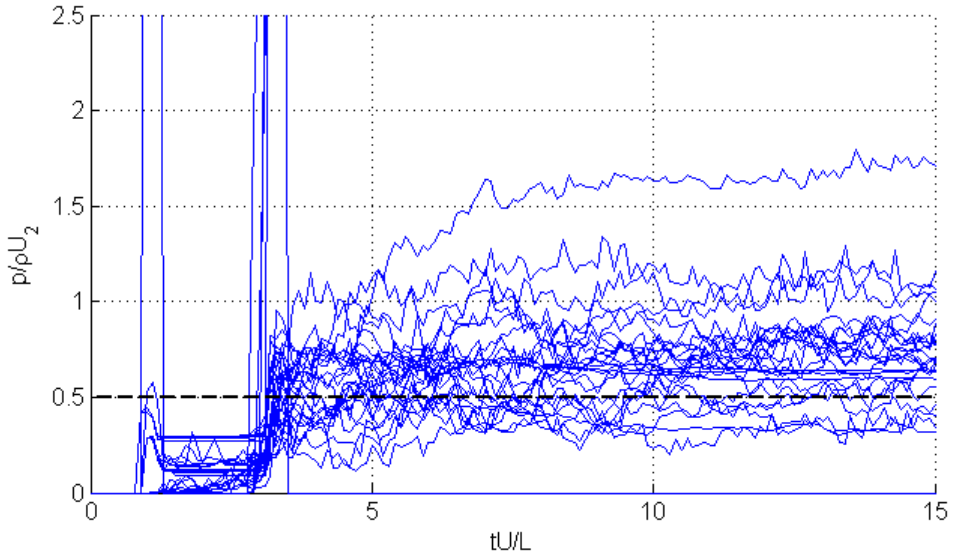


Figure 5.7.10: Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 10$ and the dashed line is the theoretical pressure.

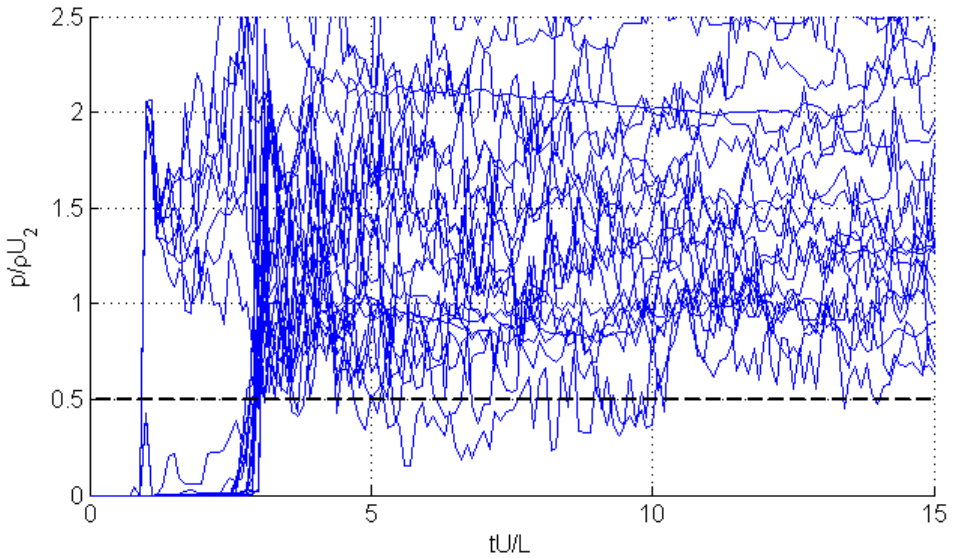


Figure 5.7.11: Dimensionless stagnation pressure over time, the blue line is $c_0/U_0 \simeq 20$ and the dashed line is the theoretical pressure.

The ratio between the jet diameter and initial particle space is evaluated for all simulation, in figure 5.7.12 this ratio is evaluated for $c_0/U_0 \simeq 5$.

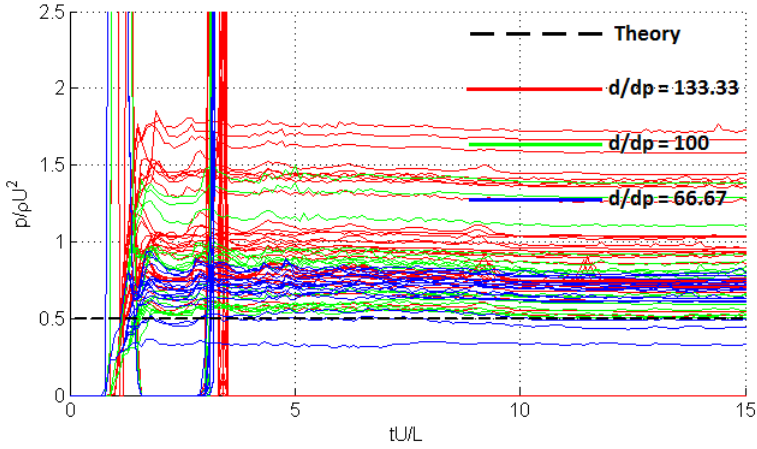


Figure 5.7.12: Dimensionless stagnation pressure over time, $c_0/U_0 \simeq 5$ and different particle space.

The results showed that a ratio of 100:1 between the water jet and initial particle space is most reasonable.

The artificial viscosity coefficient (α) is also evaluated, but there is no clear correlation between theoretical pressure and chosen coefficient.

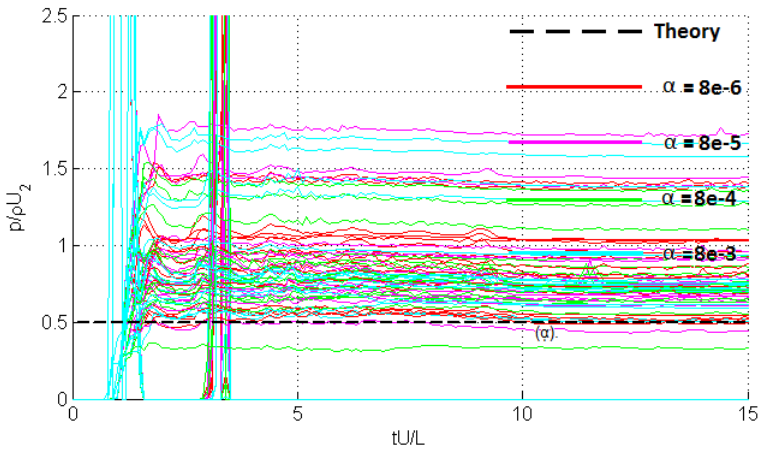


Figure 5.7.13: Dimensionless stagnation pressure over time, $c_0/U_0 \simeq 5$ and different viscosity coefficient.

The density at the boundaries is updated after a number of steps. In figure 5.7.14 the density is updated after 1, 50 and 100 time steps.

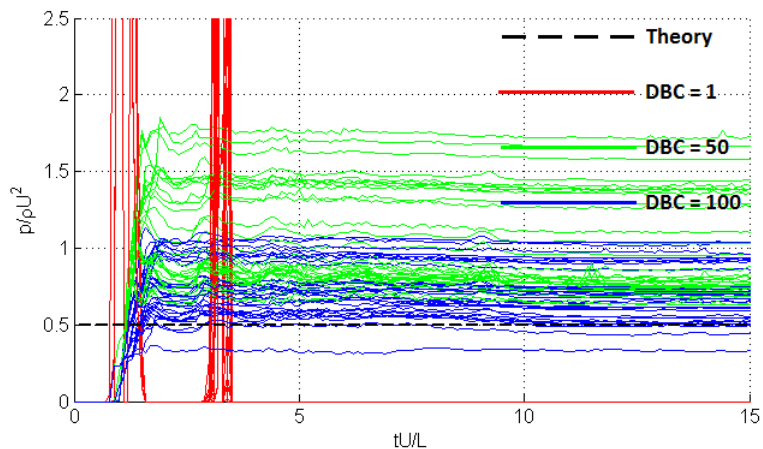


Figure 5.7.14: Dimensionless stagnation pressure over time, $c_0/U_0 \simeq 5$ and different number of steps to update the density at the plate.

The best simulation result is from simulation number 233, with 8.5 % average absolute deviation in percent from the theoretical stagnation pressure. The stagnation pressure over time from this simulation is presented in figure 5.7.15.

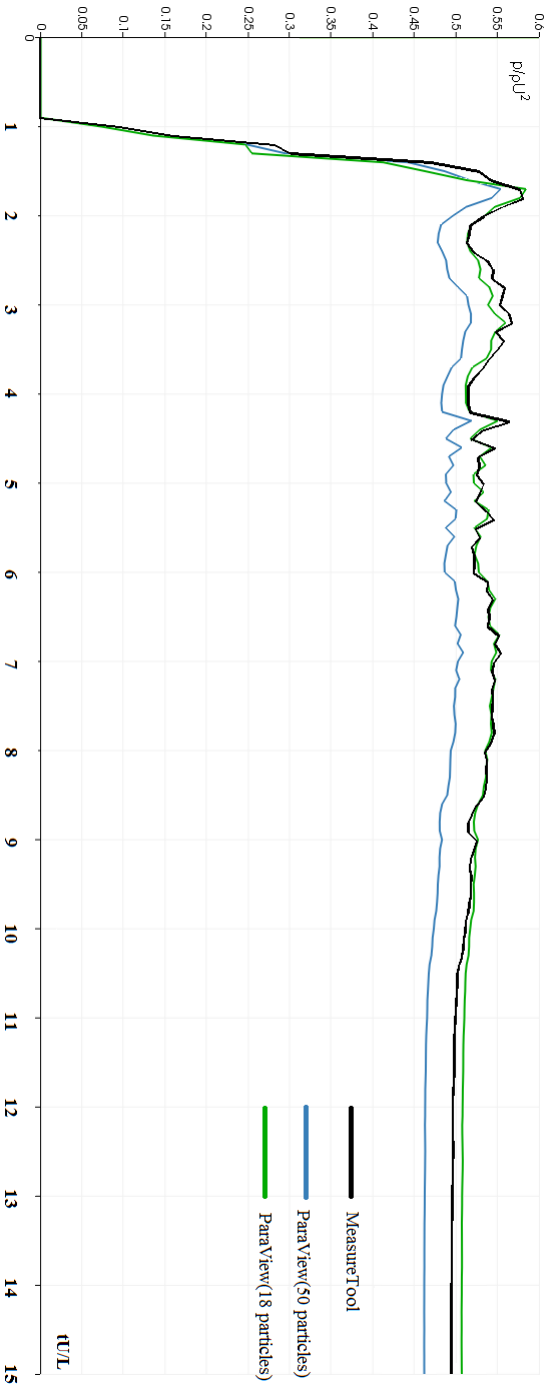


Figure 5.7.15: Jet impinging a plate. Evaluated with MeasureTool and mean values around the stagnation point from ParaView (18 particles and 50 particles).

5.7.3 Diffusion

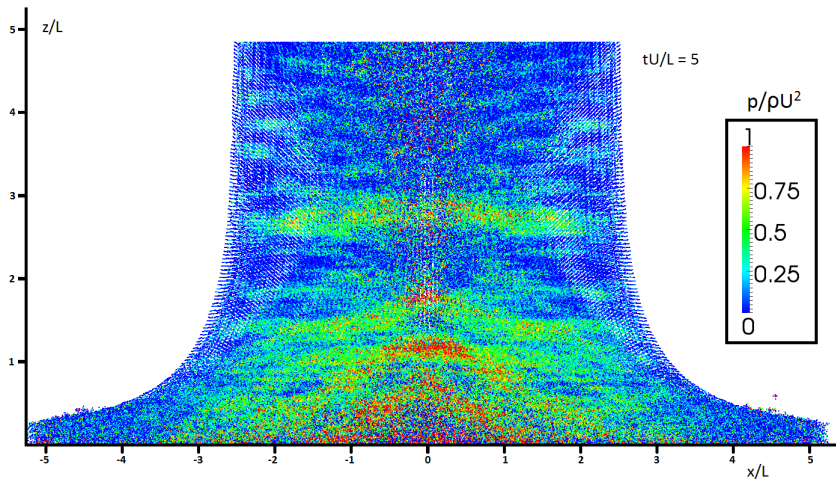


Figure 5.7.16: Enlarged view of the particles distribution close to the stagnation point obtained without density diffusion correction. Dimensionless pressure values are shown.

The figure 5.7.16 clearly shows diffuse density within the results from DualSPHysics. Density diffusion results in a large fluctuation in pressure.

5.7.4 Repulsion

A repulsion effect was observed in some of the results, when the water jet hits the plate.

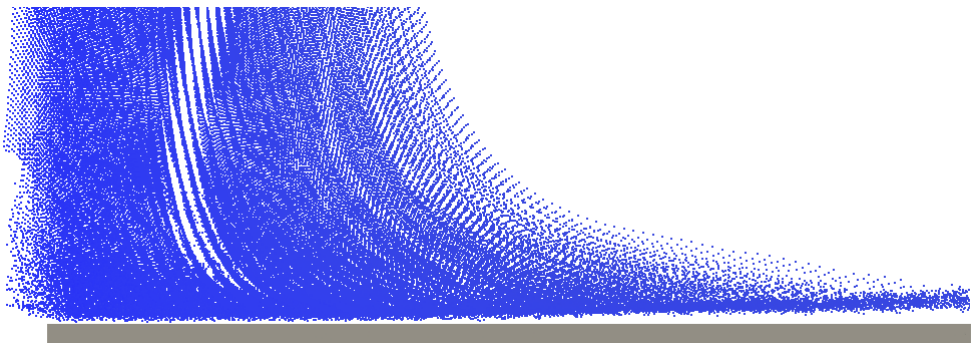


Figure 5.7.17: Repulsion at the plate.

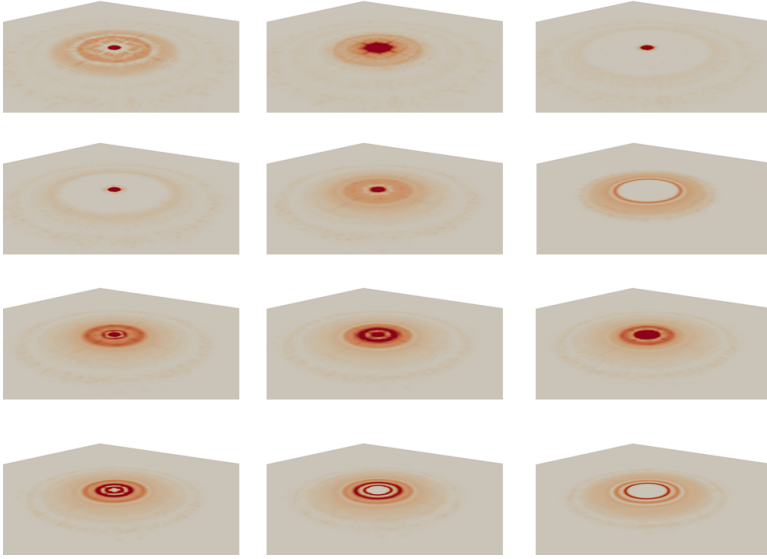


Figure 5.7.18: Illustration of the surface pressure at different time steps. The colored contours describe the pressure, a high pressure is indicated by the red color.

5.7.5 Pressure distribution

The pressure distribution over the plate have a pressure drop around one distance of a radius from the stagnation point. This is due to no air-water interaction in the simulations. Figure 5.7.19 illustrates this, and there are also large pressure oscillations in the simulations.

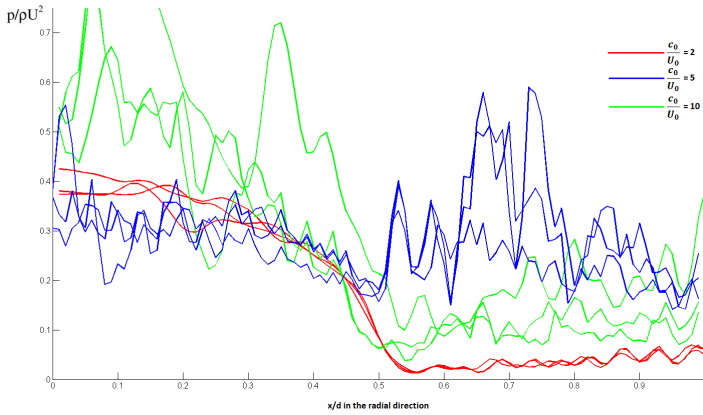


Figure 5.7.19: The pressure distribution along the plate.

5.7.6 Penetration

At $tU/L = 2$ an evaluation of number of water particles penetrating the plate was conducted.

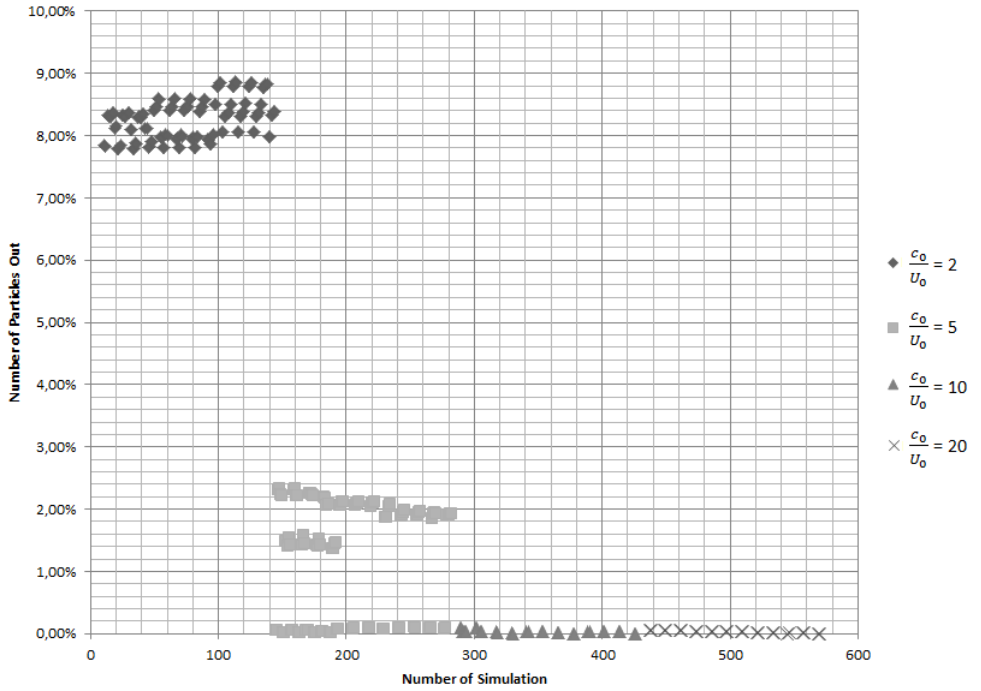


Figure 5.7.20: Percent of water particles that penetrated the plate at $tU/L = 2$.

5.7.7 Run time

The parallel programming framework and language CUDA for GPU computing appear as a good alternative to handle *High Performance Computing* (HPC) for numerical modelling. SPH simulation is very time-consuming.

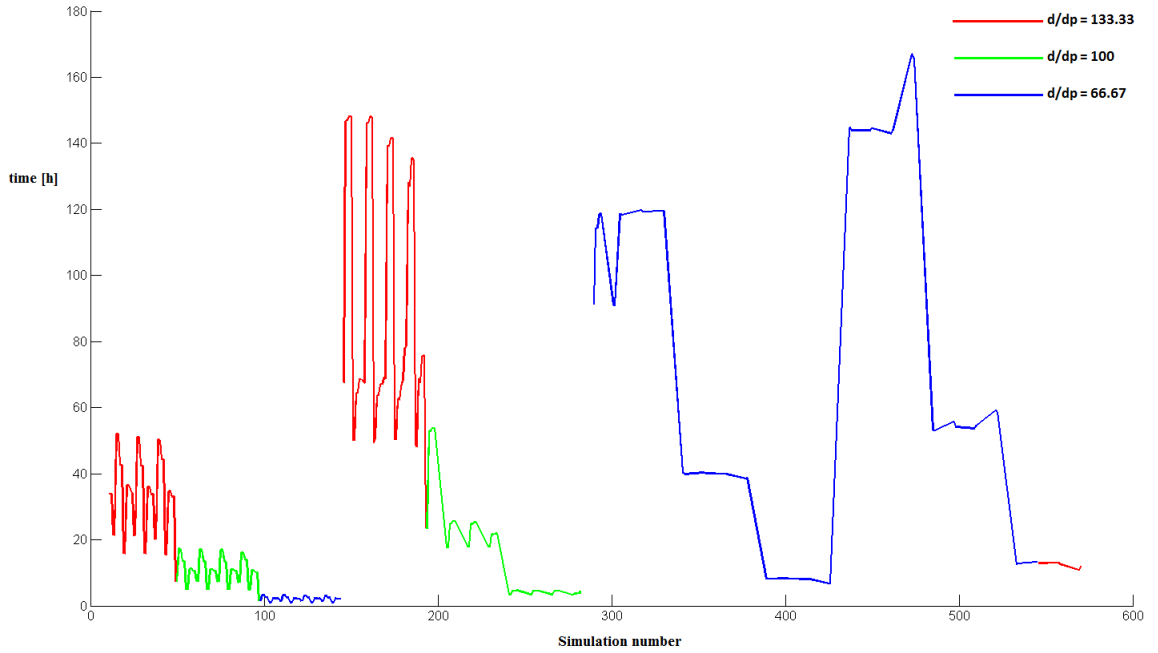


Figure 5.7.21:

Chapter 6

A Simple Pelton Bucket Geometry

The flow through a static Pelton turbine bucket has been investigated with the same geometry as used by Klemetsen [29]. The static pressure distribution over the bucket was measured with pressure taps, and pressure sweeps tangentially along the streamline at the centre of the bucket curvature was measured with a pitot probe. Numerical simulation with volume of fluid method (Fluent) and homogeneous method (CFX) was compared to the experimental data.

This case was simulated in DualSPHysics and compared with Klemetsen's results.

6.1 Pre-Processing

In DualSPHysics very complex geometries can easily be created with GenCase. Any object in STL, PLY or VTK (triangle mesh) can be converted in particles using the GenCase code. The mesh nodes that represent the selected object are store as a matrix of nodes.

6.1.1 Setup

Time algorithm	Symplectic
Viscosity formulation	Artificial
Kernel function	Wendland
Reference water density ρ_0	1000 [kg/m ³]
Uniform jet velocity U_0	20 [m/s]
Jet diameter d	0.037 [m]
Artificial speed of sound c_0	~ 250 [m/s]
Artificial viscosity coefficient α	0.1
Particle spacing	1 [mm]

Table 6.1.1: Setup variables

In the SPH-ALE simulation of Pelton turbines done by Leboeuf and Marongiu [37], the artificial speed of sound in the Tait’s equation is in the area of ten to thirteen times higher than the jet velocity.

The values of the parameters used in the simulations were chosen for three reasons, the results from the water jet impinging a plate case, recommendations from the developers of DualSPHysics and based on the values of the parameters used by Marongiu in his Pelton SPH-ALE simulations.

6.1.2 Geometry

The geometry was made of three cylinder shapes at three different origins. The bucket is quasi symmetric and the width of the bucket is equal $3.1d_j$.

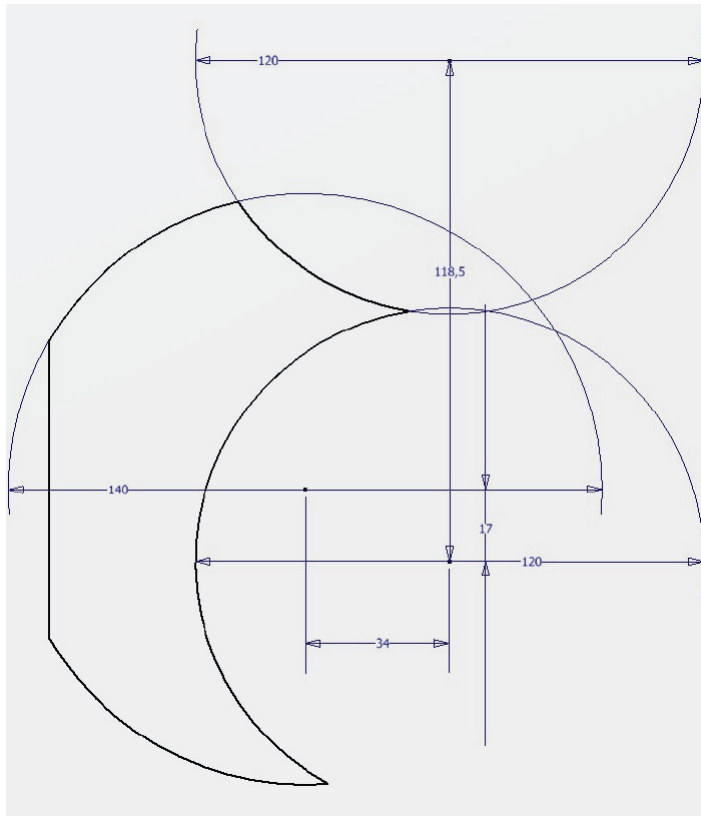


Figure 6.1.1: Schematic drawing of the "bucket", three cylinder shapes at three different origins [29].

A STL file was created in Pro/Engineer. An STL file describes only the surface geometry of a three dimensional object, and the raw unstructured triangulated surface by the unit normal and vertices of the triangles. In Pro/Engineer the values for chord height and angle control determine the triangulation mesh. In general, smaller numbers will yield a reasonable STL file for most applications.

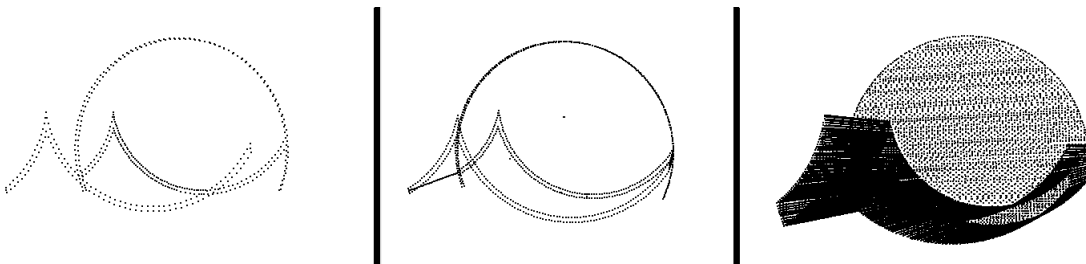


Figure 6.1.2: Point adjustment for the geometry.

Since the STL file only generates a surface of points, a method for filling the object with particles is necessary. In the initial XML file a seed point is defined inside the bucket. The seed point is bound by a box of void points and the box is larger than the size of the bucket, the bucket is then filled with bound particles.

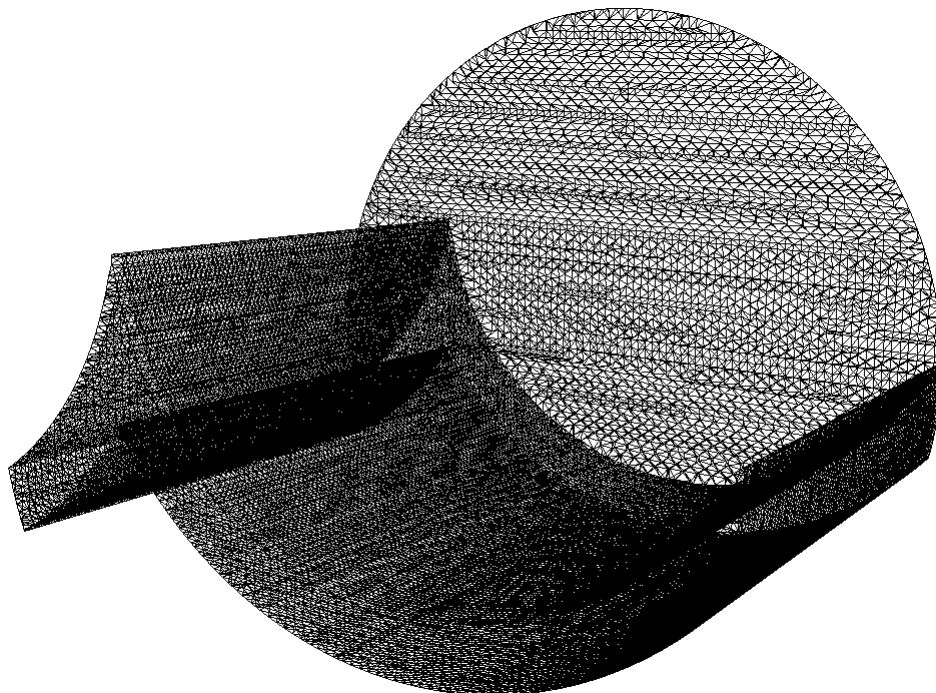


Figure 6.1.3: The final geometry for SPH simulations.

An example of implementation of seed point is shown in figure 6.1.4. The XML code fills the bucket with particles.

```

<setmkboun mk="0" />
<drawfilestl file="feit_klemetsen.stl" />
<setmkboun mk="0" />
- <fillbox x="-0.06" y="-0.03" z="0.05" mkbound="0" rem="seed point">
  <modefill>void</modefill>
  <point x="-0.087" y="-0.0705" z="-0.0005" />
  <size x="0.14" y="0.107" z="0.098" />
</fillbox>

```

Figure 6.1.4: Example code from XML file.

6.1.3 Inlet Condition

The inlet cylinder with water particles is defined by the results from pitot measurements done by Klemetsen, modelled with a number of cylinder walls with different initial velocities. The different cylinders were calculated with Matlab, where the distance between two cylinders must at least be larger than the particle spacing. The problem with this is that an accurate inlet velocity profile requires a very small particle spacing.

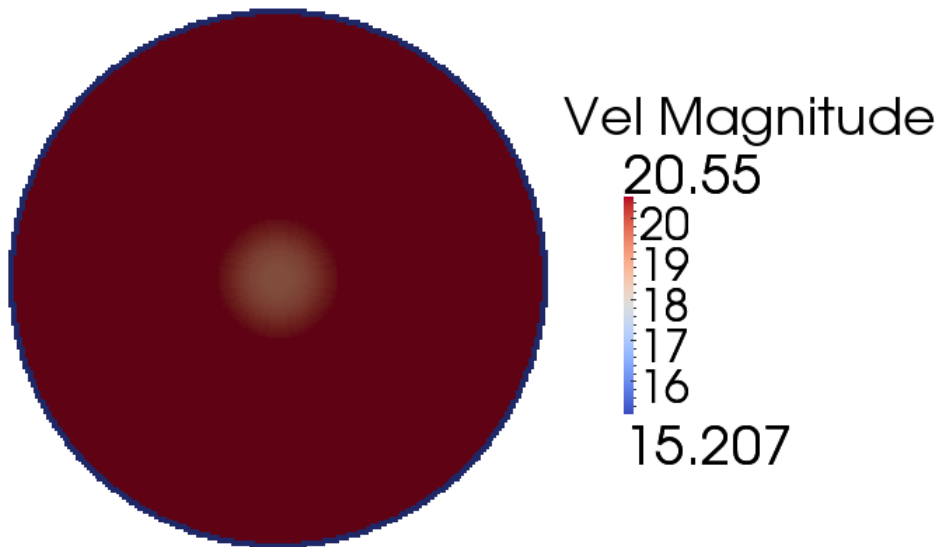


Figure 6.1.5: The generated inlet velocity profile, 100 cylinder walls were used. Plotted in ParaView.

To accurately model the wake region in the inlet condition, at least one hundred particles along the radius are needed. In this case, with a diameter of 37 mm it would require a particle spacing of 1.85×10^{-7} mm with one hundred different cylinders (200 particles along the jet), which would require a lot of memory. This implementation is therefore not used in this thesis.

6.1.4 Position Angle

The definition of the position angle at the cross section of the flow domain, the figure below can be used as reference.

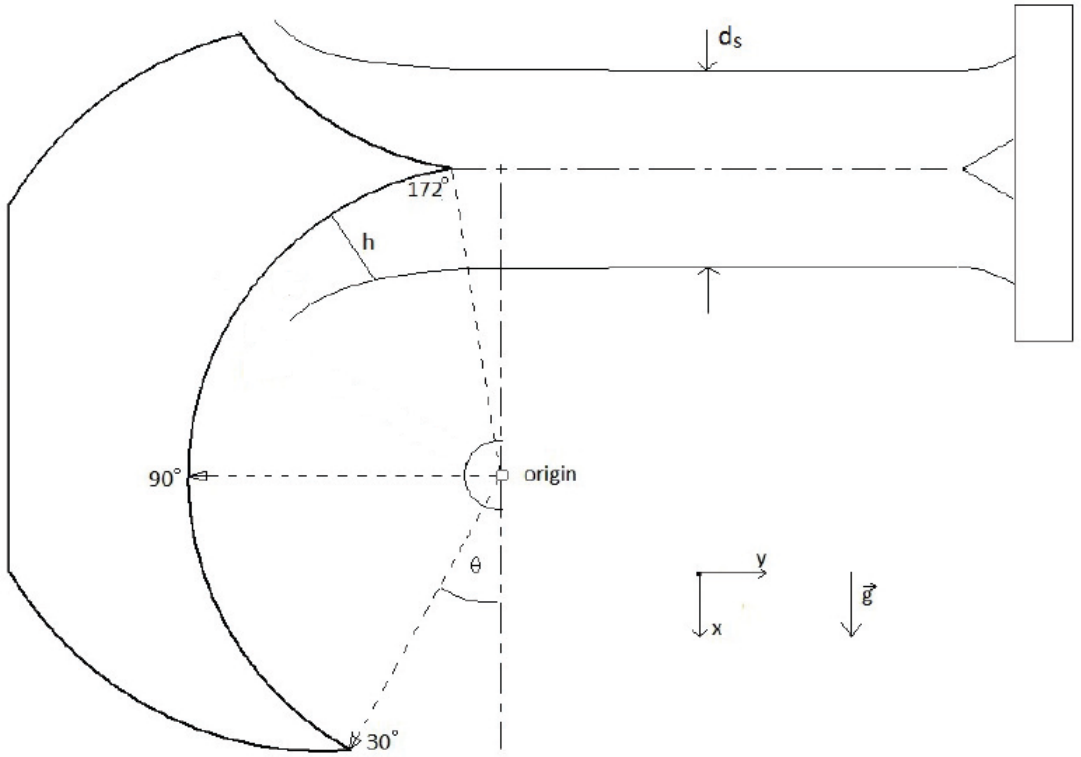


Figure 6.1.6: Cross section of the flow domain [29].

The function `atan2` is the arctangent function with two arguments. The purpose of using two arguments instead of one, is to gather information of the signs of the inputs in order to return the appropriate quadrant of the computed angle, which is not possible for the single-argument arctangent function. The position angle (θ) was defined from the following expression derived from the tangent half-angle formula in the calculator filter in ParaView, since the calculator filter does not have build-in the function `atan2`.

$$\theta = -2 \cdot \text{atan}\left(\frac{\sqrt{x^2 + y^2} - x}{y}\right) \cdot \left(\frac{180}{\pi}\right) \quad (6.1.1)$$

In Matlab, the built-in function `atan2` was used.

6.2 Post-Processing

The visualization code IsoSurface uses to extract a polygonal mesh of isosurfaces from the simulations. The isosurfaces was also processed in Blender.

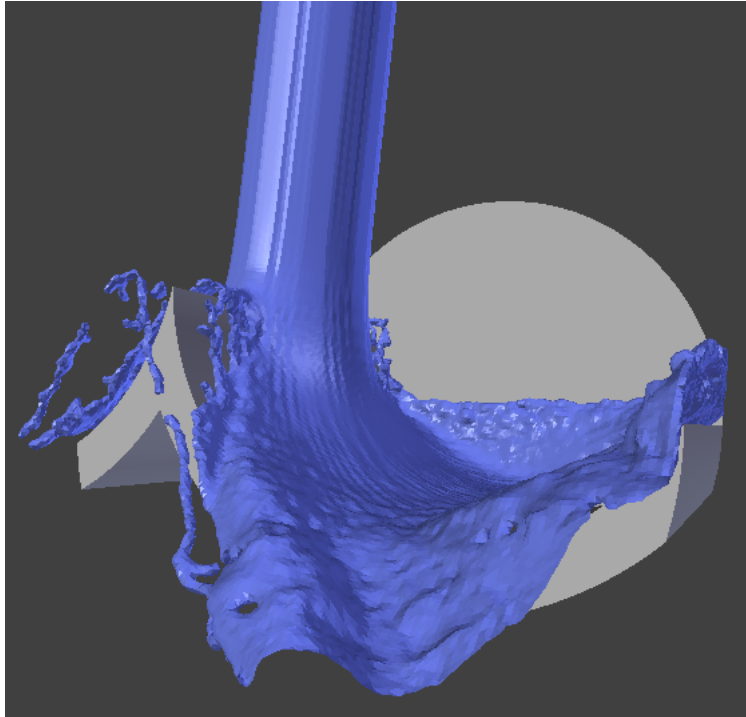


Figure 6.2.7: Isosurface of water jet impinging the bucket and processed in Blender.

6.2.1 Paraview

A sphere (with centre in $0,0,0.05$ and a radius of 0.06 [m]) was used to investigate the wall pressure at the centerline.

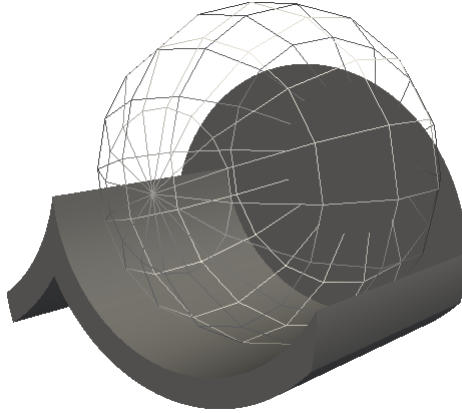


Figure 6.2.8: The clip function with the clip type sphere is used.

The particles at the centerline are then processed in Matlab.

6.2.2 MeasureTool

A plane at the centerline is created and the file with all the particles was processed in Matlab, where the water film and the wall pressure are investigated.

6.3 Validation

Simulation results of each of the simulated cases were compared to Klemetsen's results, for the validation and benchmarking of DualSPHysics.

6.4 Results

In this section the numerical results are to be presented and compared to reference values.

6.4.1 Pressure at the wall

When comparing the results from DualSPHysics to the experimental data obtain by Klemetsen the error is evident.

One possible source of error is the roughness of the bucket curve, and is illustrated in figure 6.4.10.

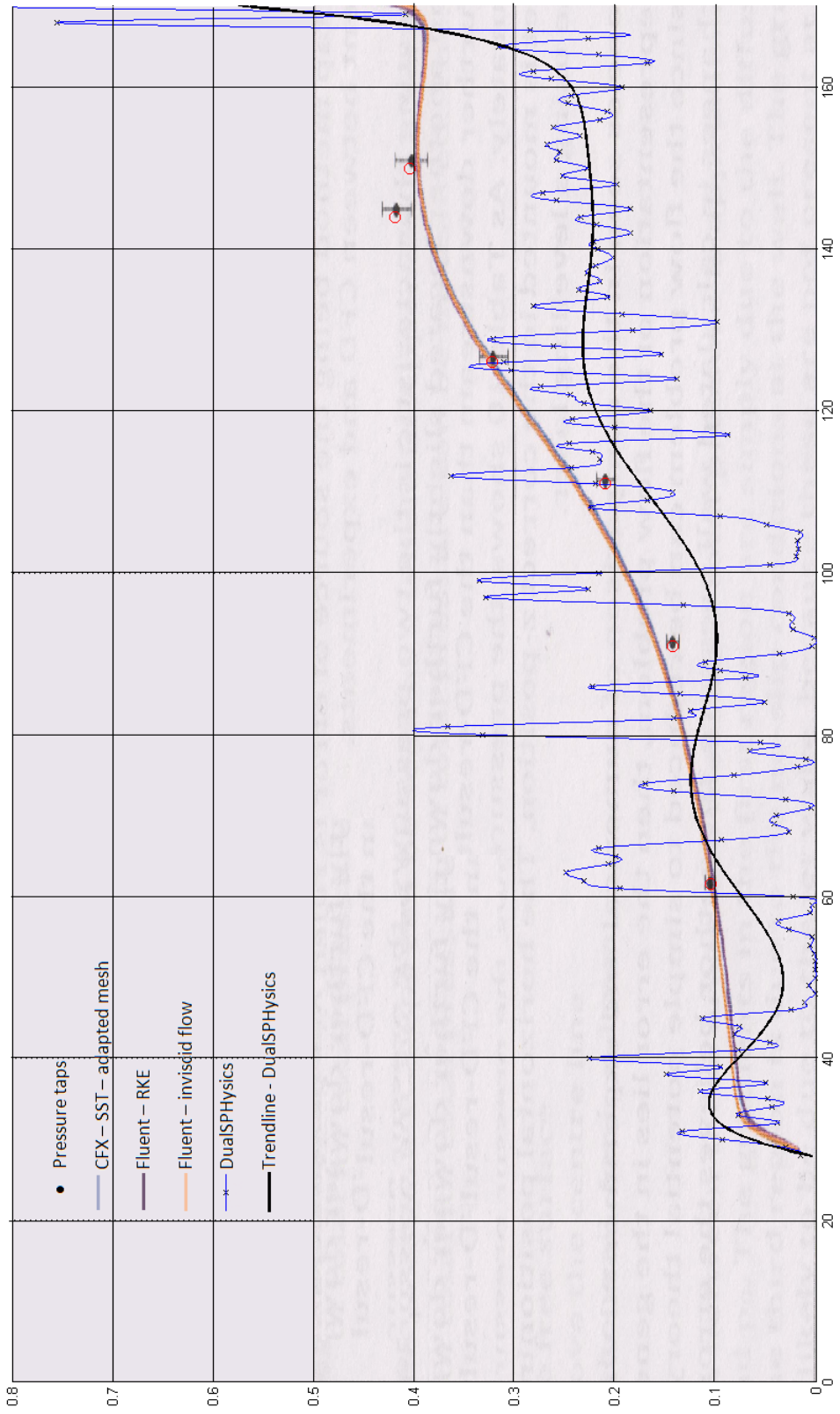


Figure 6.4.9: Comparison between experimental and numerical results for wall pressure along the centerline

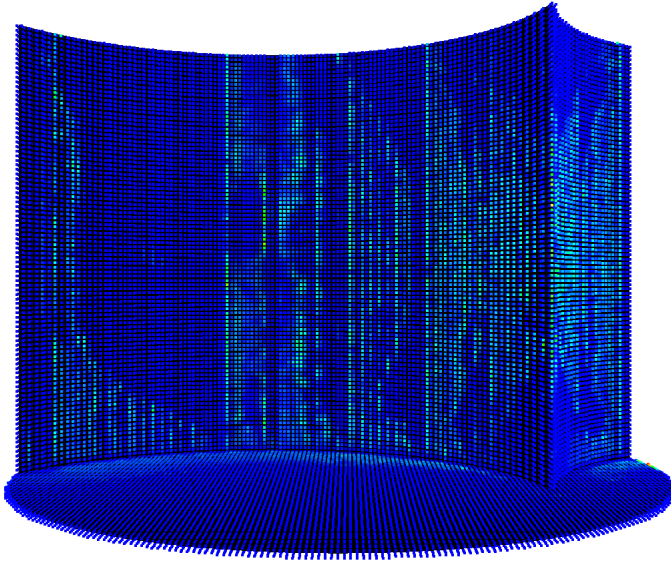


Figure 6.4.10: Pressure distribution on the bucket

6.4.2 Flow Analysis

A flow analysis is used to study the bucket flow. It has been proven as an attractive alternative to pressure analysis.

6.4.2.1 Outflow Angle

The outflow angle out of the bucket was found and compared to visual observation and numerical results from other CFD programs done by Klemetsen. The outflow angle provides information about the spreading of the impinging flow. The outflow angle is defined as the position angle the main flow is leaving the bucket.

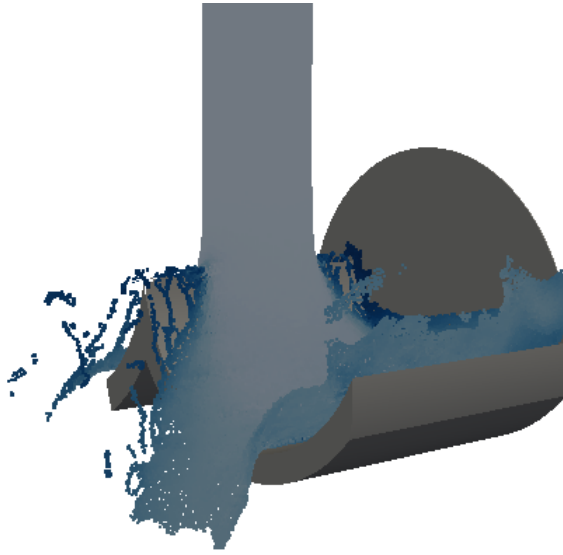


Figure 6.4.11: Outflow

Observed:	$142.4^\circ \pm 3^\circ$
Fluent	$140^\circ \pm 0.5^\circ$
CFX	$142^\circ \pm 0.5^\circ$
DualSPHysics	$139^\circ \pm 2^\circ$

Table 6.4.2: Outflow angle

The outflow angle was measured in ParaView and calculated in Matlab. The size of the estimated uncertainties related to the results are due to a few random particles going out of the bucket earlier than the main flow.

6.4.2.2 Water Film Thickness

The thickness of the water film, is measured at the centerline. The height function in MeasureTool is used to measure the thickness of the water film, h .

Due to lack of time this results were not compared to the results obtained by Klemetsen. But the results are promising.

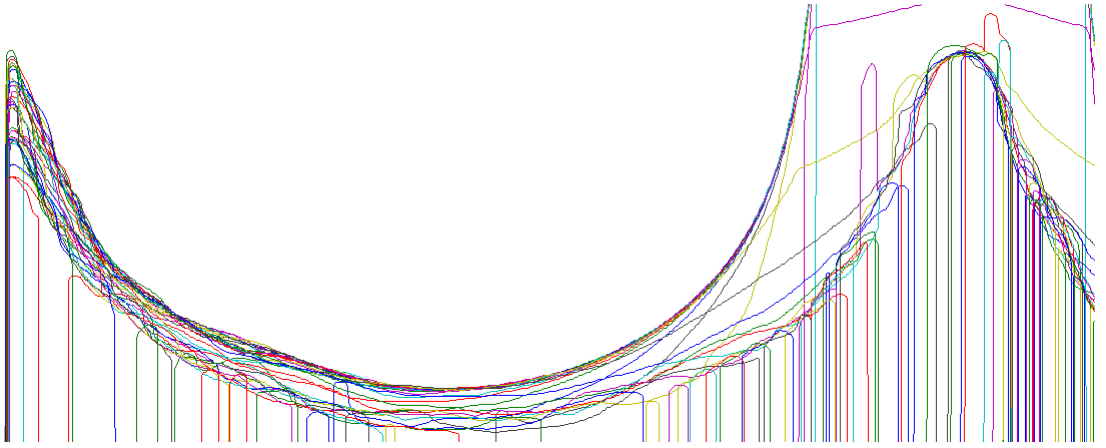


Figure 6.4.12: Water film thickness over the bucket, each line represent a time step.

Chapter 7

Discussion

The main objective of this study was to investigate the possibilities for usage of DualSPHysics as a CFD tool to predict the torque applied to a Pelton bucket, subject to a high-speed water jet. The final goal is to run a water jet on a complete Pelton runner rotating in its casing.

7.1 Accuracy

One of the main objectives of this project was to validate DualSPHysics against analytical and experimental results.

In the case of the water jet impinging the plate, the stagnation pressure at the plate is consistent with the analytical incompressible stagnation pressure at steady state in some of the simulations. But the accuracy depends heavily on the selection of the parameters and the parameter values.

The collision between the boundaries and the water particles creates a pressure increase, the water can only escape by expanding at the free surface. This is visualized well by DualSPHysics in ParaView. Yet, the results from simulations show that pressure at the stagnation point is too large (according to theory) in some of the simulations.

Due to the discrepancy between different pressure results in various simulations, the accuracy of the simulations is associated with considerable uncertainty.

7.2 Consistency

A numerical method is said to be consistent if the discretization becomes exact as the grid spacing (particle spacing for SPH) tends toward zero [18].

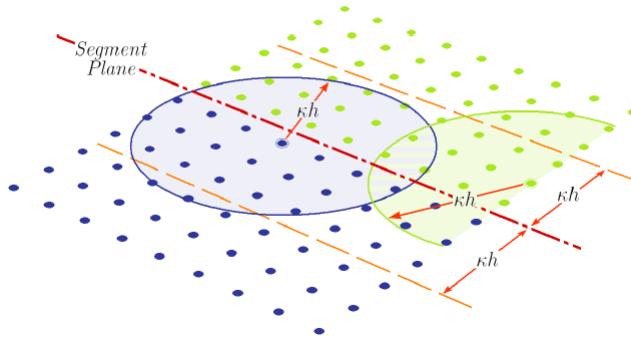


Figure 7.2.1: Problem domain with SPH particle interactions across some segmenting plane. [25]

With a smaller particle spacing the particles within the support domain of a particle will increase. It is important to ensure that an adequate number of particles reside in each support domain to ensure the error is small. In DualSPHysics the initial relationship between the support domain and particle spacing is constant.

There are two problems with reducing the particle spacing in DualSPHysics. The first is the time integration, the smaller the particle spacing is, the smaller the time step must be to ensure stability (CFL condition). The second problem is computer memory.

The results of this study clearly state a relation between accuracy and particle spacing. A lower d_j/dp ratio gave a more accurate solution.

7.3 Pressure Analysis

In this study, there are evident effects of repulsion and diffusion in the pressure solution and hence related inaccuracies. A water jet impinging normal on an object can induce very large pressure gradients, and hence a good pressure analysis is hard with SPH. Since the pressure term in the momentum equation is calculated from Tait's equation and related to an artificial compressible coefficient. This will lead to prohibitive time step that are extremely small.

In the δ -SPH a proper artificial diffusive term is used the continuity equation in order to remove the spurious numerical high-frequency oscillations in the pressure field. This will be implemented in the new version of DualSPHysics.

A projection scheme to compute incompressible flows using SPH, can also increase the pressure accuracy. Research indicates that the projection method may be exactly transcribed within the discrete context.

7.4 Flow Analysis

In general the flow looks quite good, and the flow analysis are in accordance with experimental data.

A use of flow analysis to obtain the correct streaklines of the various particles, and analysis of the momentum conservation along these streaklines will provide a partially energy transfer survey in the bucket.

7.5 Incompressibility

Incompressibility has been mentioned several times throughout this thesis. Incompressibility can be difficult to model accurately in the SPH framework. Incompressibility is not only of visual importance, in situations where a large quantity of compressible fluid is constrained the pressure can become extremely high on a small area. In the simulations done in DualSPHysics the outputs never grow without bounds, but some of the numerical results do not comply with real world observations.

In the last few years the noisy dissipation from weakly compressible formulations are being studied. Within the SPHERIC community robust incompressible SPH schemes are under development.

As the water is modelled as a weakly compressible fluid, a very small time step is required to fulfill the CFL condition.

7.6 Choosing Parameters and Functions

The combination of parameters and functions have been chosen based on advice from the developers of DualSPHysics and experience gained during this study. The parameters are corrected according to both the visual output and the incompressible stagnation pressure.

Results from simulations especially shows the importance of choosing a good C_{coef} and hence an artificial speed of sound, and adjust it accordingly to the velocity of the jet.

7.6.1 Artificial Speed of Sound

In high Mach number flows the penetration problem is severe, as described in section 3.5.10.7. The artificial speed of sound is recommend to have a limit of a $Ma_r \sim 0.1$. With a reference Mach number higher than 0.1 ($\frac{c_0}{U_0} < 10$), there is a great risk of penetration.

In this study a reasonable artificial speed of sound is five to fifteen times the initial velocity, which is in line with current theory on the topic.

Results from this study show that the artificial speed of sound depend on the simulated case, and the relationship between jet velocity and artificial speed of sound does not scale linearly, which is unfortunate.

The dimensionless Froude number (Fr) can be a useful classification, describing the ratio of a characteristic velocity to a gravitational wave velocity. $Fr = \frac{U_0 c_0^2}{g D_c}$, where D_c is defined as the hydraulic depth, and defined as h_{SWL} in DualSPHysics.

A reasonable assumption of C_{coef} can therefore be to allow for an appropriate wave velocity, and ensure the density variation is within a small limit.

7.7 Viscosity

In this study an artificial viscosity model is used. The model uses empirical coefficient to model the energy dissipation. The artificial viscosity model implemented in DualSPHysics is very often used in SPH due to its simplicity. But alas it is empirical.

In this study a good artificial speed of sound seems more important for an accurate solution than the artificial viscosity coefficient.

7.8 Boundary Treatment

As shown in section 5.7.4, the repulsion affects the pressure results significantly. This study had success with the use of not updating the pressure values of the boundary particles at each time step in order to minimize repulsion effects. According to Dr. Alejandro J.C. Crespo [13] this is not a viable option in boundary treatment, because numerical solutions are to be preferred. In futher versions of DualSPHysics a new boundary treatment model is going to be implemented.

Especially in the SPH-ALE boundary treatment framework, the boundary treatment can be much more accurately than in the original SPH framework.

Chapter 8

Conclusion

The understanding of issues related to the stability, accuracy and convergence properties of the SPH method are gradually increasing. According to Liu and Liu in 2003, [32] "there is still a long way for the method to become extensively applicable, practically useful and robust as the traditional grid-based methods such as FVM and FDM. This is because much work needs to be done to consolidate the theoretical foundations of the SPH method." After 2005 and the founding of the SPHERIC [1], the SPH method have undergone a large development. But in a CFD framework the method is still in its childhood.

The development of the DualSPHysics solver made the SPH method more manageable and available. The work in this thesis shows some of the advantages and disadvantages of DualSPHysics and the SPH method. Even though an uncomplicated uniform flow is studied, the accuracy of the simulations is associated with considerable uncertainty. Not even the best simulation showed satisfying accuracy. The two case studies in this thesis show how small alterations to the program can change the solution.

Scientific validation is crucial in the work of an engineer, and in order to validate DualSPHysics for Pelton bucket simulations much work remains. But the SPH community is innovative, and SPH is likely to become a key method for free surface flow simulation in the future.

Challenges found in this study includes repulsion effects and pressure oscillation. It is true that the boundary treatment is the worst culprit of problems for free surface flow in DualSPHysics.

In summary, the SPH method is in fact a good choice when modelling fluid flow problems with complex meshes in motion. But it is necessary to combine the right parameters and functions to get a good solution.

Chapter 9

Further Work

The results presented in this thesis indicate that the future for SPH computational simulations in Pelton turbines is promising. Further steps in the investigation of usage of DualSPHysics simulations of flow on a Pelton turbine include more flow analysis and pressure analysis.

A new version of DualSPHysics is announced to be released by July 2013, the version 3.0. The new version with several new improvement may solve some of the problems experienced with DualSPHysics in this study.

9.1 Further Comparison to Previous Studies

Further investigation of comparison between DualSPHysics and previous studies should be carried out in order to increase the understanding of the SPH method and the implementation of DualSPHysics. Thus increase the accuracy of simulations in DualSPHysics.

9.2 Inlet Conditions

DualSPHysics is planning to implement inlet conditions in the future. In terms of the source code, the values of N_p , N_{fluid} , will not longer be constant. The inlet condition implemented by Marongiu [39] [38] is not properly explained in his papers. He created disks of particles with a given gradient of pressure and desired velocity, and each new disk creating and forming the jet.

9.3 Parameters in DualSPHysics

The different parameters in DualSPHysics are according to this study very case sensitive, and further investigation may increase the accuracy. Nowadays, there is a try and fail methodology within DualSPHysics community, and new and more accurate methods are being implemented.

9.4 Equipment for Counteracting Effect of Repulsion

A suggestion by Dr. Benedict D. Rogers is to have a cylinder with a lid around the water jet for counteracting effect of repulsion.

9.4.1 Flow Analysis

Anagnostopoulos and Papantonis [2] suggest a flow analysis for calculation of the mechanical torque produced by a Pelton runner from the conservation of angular momentum.

The author suggest simulation with one single Pelton bucket with circular motion and a jet of water impinging tangentially on to the bucket, in order to try to use this approach to calculate mechanical torque and hydraulic efficiency.

Bibliography

- [1] Spheric newsletter - sph european research interest community, 2005-2012.
- [2] J. S. Anagnostopoulos and D. E. Papantonis. A fast lagrangian simulation method for flow analysis and runner design in pelton turbines. *Journal of hydrodynamics*, 24(6):930–941, 2012.
- [3] Rutherford Aris. *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Dover Publications, 1990.
- [4] D.J. Auld and K. Srinivas. *Aerospace*, 1995.
- [5] L. F. Barstad. Cfd-analysis of a pelton turbine. Master’s thesis, Norwegian University of Science and Technology, 2012.
- [6] S. Beltaos and N. Rajaratnam. Impingement of axisymmetric developing jets. *Journal of Hydraulic Research*, 15:4:311–326, 1977.
- [7] H. Brekke. *Pumper & Turbiner*. Vannkraftlaboratoriet NTNU, 2003.
- [8] H. Chanson. *Fluid Mechanics of Environmental Interfaces*. Taylor & Francis Books Ltd., 2012.
- [9] A. Colagrossi. *A Meshless Lagrangian Method for Free Surface and Interface Flows with Fragmentation*. PhD thesis, Università di Roma, 2003.
- [10] A. Colagrossi and et.al. Theory and applications of delta-sph and m-sph models. Course for NTNU PhD students - 5 May 2011.
- [11] A. C. Crespo and et.al. Gpus, a new tool of acceleration in cfd: Efficiency and reliability on smoothed particle hydrodynamics methods. *PLoS ONE*, 6, Issue 6, e20685, 2011.
- [12] A. J. C. Crespo. *Application of the Smoothed Particle Hydrodynamics model SPHysics to free-surface hydrodynamics*. PhD thesis, Universidade de Vigo, 2008.
- [13] A. J.C. Crespo. Personal correspondence, 2013.
- [14] A.J.C. Crespo and et.al. *User Guide for DualSPHysics code*. SPHysics project, v2.0 edition, March 2012.

- [15] R.A. Dalrymple and B.D. Rogers. Numerical modeling of water waves with the sph method. *Coastal Engineering*, 53:141–147, 2006.
- [16] J.M. Dominguez and et.al. Sphysics project, August 2007.
- [17] R. Fatehi and M. T. Manzari. A remedy for numerical oscillations in weakly compressible smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 67:1100–1114, 2011.
- [18] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 1999.
- [19] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.
- [20] M. Gomez-Gesteira and et.al. Sphysics - development of a free-surface fluid solver - part 1: Theory and formulations. *Computers & Geosciences*, 48:289–299, 2012.
- [21] M. Gomez-Gesteira and et.al. Sphysics - development of a free-surface fluid solver - part 2: Efficiency and test cases. *Computers & Geosciences*, 48:300–307, 2012.
- [22] M. G. Gomez-Gesteira and et.al. State-of-the-art of classical sph for free-surface flows. *Journal of Hydraulic Research*, Vol. 48:6–27, 2010.
- [23] V. Gupta and V. Prasad. Numerical investigations for jet flow characteristics on pelton turbine bucket. *International Journal of Emerging Technology and Advanced Engineering*, 2:364–370, 2012.
- [24] M.P. Hansen. Study of the sph method and using a 2-d model of a wave flume. Master’s thesis, Aalborg University, 2008.
- [25] Williamsb J. R. Holmes, D. W. and P. Tilkec. A framework for parallel computational physics algorithms on multi-core: Sph in parallel. *Advances in Engineering Software*, 2009.
- [26] T. Karimipannah and H.B. Awbi. Theoretical and experimental investigation of impinging jet ventilation and comparison with wall displacement ventilation. *Building and Environment*, 37:1329–1342, 2002.
- [27] M. Kelager. Lagrangian fluid dynamics using smoothed particle hydrodynamics. Master’s thesis, University of Copenhagen, 2006.
- [28] A. Khayyer and H. Gotoh. Enhancement of stability and accuracy of the moving particle semi-implicit method. *Journal of Computational Physics*, 230:3093–3118, 2011.
- [29] L.E. Klemetsen. An experimental and numerical study of the free surface pelton bucket flow. Master’s thesis, NTNU, 2010.

- [30] Anagnostopoulos J. S. Koukouvini, P. K. and D. E. Papantoni. Flow modelling in the injector of a pelton turbine. In *4th SPHERIC Workshop*, 2009.
- [31] J. et. al Leduc. A sph-ale method to mode multiphase flows with surface tension. In *7th International Conference on Multiphase Flow, ICMF, Tampa, FL, May 30-June 4, 2010*.
- [32] M.B. Liu and G. R. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing Co Pte Ltd, 2003.
- [33] M.B. Liu and G.R. Liu. Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics*, 56:19–36, 2006.
- [34] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astrophysical Journal*, 82:1013–1024, 1977.
- [35] Black M. and J. King. *The Atlas of Water*. University of California Press, 2009.
- [36] Hamid Maghzian. *Simulation of Hydrodynamics of the Jet Impingement using Arbitrary Lagrangian Eulerian Formulation*. PhD thesis, The University of British Columbia, 2007.
- [37] Frédéric Magoulès, editor. *Computational Fluid Dynamics*. Chapman & Hall/CRC, 2011.
- [38] J.C. Marongiu and et.al. Free surface flows simulations in pelton turbines using an hybrid sph-ale method. *Journal of Hydraulic Research*, 48:40–49, 2010.
- [39] Leboeuf F. Marongiu, J.C. and E. Parkinson. Numerical simulation of the flow in a pelton turbine using the meshless method smoothed particle hydrodynamics: a new simple solid boundary treatment. *Journal of Power and Energy*, 221:849–856, 2007.
- [40] D. Molteni and A. Colagrossi. A simple procedure to improve the pressure evaluation in hydrodynamic context using the sph. *Computer P*, 180:861–872, 2009.
- [41] J.J. Monaghan. On the problem of penetration in particle methods. *Journal of computational physics*, 82:1–15, 1989.
- [42] J.J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–74, 1992.
- [43] J.J Monaghan and A. Kos. Solitary waves on a cretan beach. *Journal of waterway, port, coastal and ocean engineering*, 125:145–155, 1999.
- [44] Kubota T. Nakanishi, Y. and T. Shin. Numerical simulation of flows on pelton buckets by particle method. In *Proceedings of the XXI IAHR Symposium Hydraulic Machinery and Systems Volume I*, 2002.

- [45] W. L. Oberkampf and T. G. Trucano. Verification and validation in computational fluid dynamics. Technical report, Sandia National Laboratories, 2002.
- [46] A. Perrig. *Hydrodynamics of the free surface flow in pelton turbine buckets*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2007.
- [47] N. Rajaratnam. *Turbulent Jets*. Elsevier Scientific Publishing Company, 1976.
- [48] J. Ren and et.al. Simulation of container filling process with two inlets by improved smoothed particle hydrodynamics (sph) method. *International Journal of Computational Fluid Dynamics*, 25:365–386, 2011.
- [49] H.K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics*. Pearson, 2007.
- [50] D. Violeau. *Fluid Mechanics and the SPH Method*. Oxford University Press, 2012.
- [51] M. F. White. *Fluid Mechanics*. McGraw Hill, 2008.

Appendix A

DualSPHysics XML file example

```
<?xml version="1.0" encoding="UTF-8" ?>
<case app="GenCase v1.3 (14-02-2012)" date="18-05-2013 15:39:19">
  <casedef>
    <constantsdef>
      <lattice bound="1" fluid="1" />
      <gravity x="0" y="0" z="-9.81" />
      <cflnumber value="0.2" />
      <hswl value="0" auto="true" />
      <coefsound value="400" />
      <coefficient value="0.866025" />
      <gamma value="7" />
      <rhop0 value="1000" />
      <eps value="0.5" />
    </constantsdef>
    <mkconfig boundcount="240" fluidcount="10">
      <mkorientfluid mk="0" orient="Xyz" />
    </mkconfig>
    <geometry>
      <definition dp="0.001">
        <pointmin x="-0.09" y="-0.08" z="-0.001" />
        <pointmax x="0.06" y="0.6" z="0.1" />
      </definition>
      <commands>
        <mainlist>
          <setshapemode>real | dp | bound</setshapemode>
          <setdrawmode mode="full" />
          <setmkbound mk="0" />
          <drawfilestl file="feit.klemetsen.stl" />
          <setmkbound mk="0" />
          <fillbox x="-0.06" y="-0.03" z="0.05" mkbound="0" rem="seed point">
            <modefill>void</modefill>
            <point x="-0.087" y="-0.0705" z="-0.0005" />
          </fillbox>
        </mainlist>
      </commands>
    </geometry>
  </casedef>
</case>
```

```

        <size x="0.14" y="0.107" z="0.098" />
    </fillbox>
    <shapeout file="" />
    <setmkfluid mk="0" />
    <drawcylinder radius="0.0185">
        <point x="-0.05925" y="0.1" z="0.05" />
        <point x="-0.05925" y="0.6" z="0.05" />
    </drawcylinder>
</mainlist>
</commands>
</geometry>
<initials>
    <velocity mkfluid="0" x="0.0" y="-20.0" z="0" />
</initials>
</casedef>
<execution>
    <parameters>
        <parameter key="StepAlgorithm" value="2" comment="Step Algorithm 1:Verlet, 2:Symplectic" />
        <parameter key="VerletSteps" value="50" comment="Verlet only: Number of steps to take" />
        <parameter key="Kernel" value="2" comment="Interaction Kernel 1:Cubic Spline, 2:Wendland" />
        <parameter key="KernelGradientCorr" value="0" comment="Apply Kernel Gradient Correction" />
        <parameter key="ViscoTreatment" value="1" comment="Viscosity Formulation 1:Artificial, 2:Stokes" />
        <parameter key="Visco" value="0.1" comment="Viscosity value" />
        <parameter key="ShepardSteps" value="0" comment="Number of steps to apply Shepard interpolation" />
        <parameter key="DBCSteps" value="1" comment="Number of steps to update the density" />
        <parameter key="DtIni" value="1e-5" comment="Initial time step" />
        <parameter key="DtMin" value="1e-6" comment="Minimum time step (def=0.00001)" />
        <parameter key="TimeMax" value="0.05" comment="Time of simulation" />
        <parameter key="TimeOut" value="0.001" comment="Time between output files" />
        <parameter key="IncZ" value="0.5" comment="Increase of Z+" />
        <parameter key="PartsOutMax" value="1" comment="Allowed percentage of fluid particles to be output" />
    </parameters>
</execution>
</case>

```

Appendix B

Matlab Script

B.1 Run DualSPHysics Script

Matlab script for automatically run of a number of simulations.

```
clear all;
clc;

%Change directory:

gencase='C:\kjartafu\DualSPHysics_v2.01_windows_64bit\EXECS\GenCase_win64.exe';
dualsphysics='C:\kjartafu\DualSPHysics_v2.01_windows_64bit\EXECS\DualSPHysics_win64.exe';
boundaryvtk='C:\kjartafu\DualSPHysics_v2.01_windows_64bit\EXECS\BoundaryVTK_win64.exe';
partvtk='C:\kjartafu\DualSPHysics_v2.01_windows_64bit\EXEC\PartVTK_win64.exe';
measuretool='C:\kjartafu\DualSPHysics_v2.01_windows_64bit\EXECS\MeasureTool_win64.exe';
isosurface='C:\kjartafu\DualSPHysics_v2.01_windows_64bit\EXECS\IsoSurface_win64.exe';

parametersMatrix = zeros(5,6);
labels = {'Ccoef', 'dp', 'alpha', 'Shepard', 'DBC', 'timeStep'};
% TimeStepAlg = Symplectic (2)
% Kernel = Wendland (2)
% ViscoTreat = Artificial Viscosity (1)
% Velocity = 40 m/s

%Parameters
%=====

%C_coef
parametersMatrix(:,1)=[40,100,200,400,0]; %4
%dp
parametersMatrix(:,2)=[0.0015,0.002,0.003,0,0]; %3
%alpha
parametersMatrix(:,3)=[1/125000,1/12500,1/1250,1/125,0]; %4
%Shepard
```

```

parametersMatrix(:,4)=[0,50,0,0,0]; %2
%DBC
parametersMatrix(:,5)=[1,50,100,0,0]; %3
%dtMin
parametersMatrix(:,6)=[0.000000005,0.00000001,0,0,0]; %2

nos = 4*3*4*2*3*2;
fprintf('Number of simulations = %d',nos);

%Create "RUN" matrix
%=====
runMatrix=zeros(nos,7);
k=1;
for ii=1:4
    for ij=1:3
        for ik=1:4
            for jj=1:2
                for jk=1:3
                    for kk=1:2
                        runMatrix(k,1)=k;
                        runMatrix(k,2)= parametersMatrix(ii,1);
                        runMatrix(k,3)= parametersMatrix(ij,2);
                        runMatrix(k,4)= parametersMatrix(ik,3);
                        runMatrix(k,5)= parametersMatrix(jj,4);
                        runMatrix(k,6)= parametersMatrix(jk,5);
                        runMatrix(k,7)= parametersMatrix(kk,6);
                        k=k+1;
                    end
                end
            end
        end
    end
end

%Big loop:

outMeas = sprintf('md MeasurementRes');
system(outMeas);

for k=1:length(runMatrix)

if exist('Jet.xml')
    system('del Jet.xml');
end

%Create XML files
%=====

% Write to file
system('cd. > Jet.txt') % Create a txt file

fid = fopen('Jet.txt','w'); % Open with write rights

%XML FILE LOOP
fprintf(fid, '<?xml version="1.0" encoding="UTF-8" ?>\n');

```

```

fprintf(fid, '<case app="GenCase v1.3 (14-02-2012)" date="11-11-2012 16:22:53">\n');
fprintf(fid, '  <casedef>\n');
fprintf(fid, '    <constantsdef>\n');
fprintf(fid, '      <lattice bound="1" fluid="1" />\n');
fprintf(fid, '      <gravity x="0" y="0" z="-9.81" />\n');
fprintf(fid, '      <cflnumber value="0.2" />\n');
fprintf(fid, '      <hswl value="0" auto="true" />\n');
fprintf(fid, '      <coefsound value="%d" />\n', runMatrix(k,2)); % Change value
fprintf(fid, '      <coefficient value="0.866025" />\n');
fprintf(fid, '      <gamma value="7" />\n');
fprintf(fid, '      <rhop0 value="1000" />\n');
fprintf(fid, '      <eps value="0.5" />\n');
fprintf(fid, '    </constantsdef>\n');
fprintf(fid, '    <mkconfig boundcount="240" fluidcount="10">\n');
fprintf(fid, '      <mkorientfluid mk="0" orient="Xyz" />\n');
fprintf(fid, '    </mkconfig>\n');
fprintf(fid, '    <geometry>\n');
fprintf(fid, '      <definition dp="%10.7f">\n', runMatrix(k,3)); % Change value
fprintf(fid, '        <pointmin x="-0.3" y="-0.3" z="-0.1" />\n');
fprintf(fid, '        <pointmax x="0.3" y="0.3" z="0.5" />\n');
fprintf(fid, '      </definition>\n');
fprintf(fid, '    <commands>\n');
fprintf(fid, '      <mainlist>\n');
fprintf(fid, '        <setshapemode>dp | real | bound</setshapemode>\n');
fprintf(fid, '        <setdrawmode mode="full" />\n');
fprintf(fid, '        <setmkbound mk="0"/>\n');
fprintf(fid, '        <drawbox>\n');
fprintf(fid, '          <boxfill>solid</boxfill>\n');
fprintf(fid, '          <point x="-0.2" y="-0.2" z="-0.05" />\n');
fprintf(fid, '          <size x="0.4" y="0.4" z="0.06" />\n');
fprintf(fid, '        </drawbox>\n');
fprintf(fid, '        <drawpoints>\n');
fprintf(fid, '          <point x="-0.21" y="0" z="0.01" />\n');
fprintf(fid, '          <point x="0.21" y="0" z="0.01" />\n');
fprintf(fid, '          <point x="0" y="-0.21" z="0.01" />\n');
fprintf(fid, '          <point x="0" y="0.21" z="0.01" />\n');
fprintf(fid, '        </drawpoints>\n');
fprintf(fid, '        <setmkfluid mk="0" />\n');
fprintf(fid, '        <drawcylinder radius="0.1">\n');
fprintf(fid, '          <point x="0.0" y="0.0" z="0.05" />\n');
fprintf(fid, '          <point x="0.0" y="0.0" z="0.45" />\n');
fprintf(fid, '        </drawcylinder>\n');
fprintf(fid, '      </mainlist>\n');
fprintf(fid, '    </commands>\n');
fprintf(fid, '  </geometry>\n');
fprintf(fid, '  <initials>\n');
fprintf(fid, '    <velocity mkfluid="0" x="0" y="0" z="-40.0" />\n'); % Velocity
fprintf(fid, '  </initials>\n');
fprintf(fid, '</casedef>\n');
fprintf(fid, '<execution>\n');
fprintf(fid, '  <parameters>\n');
fprintf(fid, '    <parameter key="StepAlgorithm" value="2" comment="Step Algorithm">\n');
fprintf(fid, '    <parameter key="VerletSteps" value="50" comment="Verlet only">\n');
fprintf(fid, '    <parameter key="Kernel" value="2" comment="Interaction Kernel">\n');
fprintf(fid, '    <parameter key="KernelGradientCorr" value="1" comment="Apply Kernel Gradient Correction">\n');
fprintf(fid, '    <parameter key="ViscoTreatment" value="1" comment="Viscosity Treatment">\n');
fprintf(fid, '    <parameter key="Visco" value="%15.10f" comment="Viscosity value">\n', runMatrix(k,4));

```

```

fprintf(fid, '                <parameter key="ShepardSteps" value="%d" comment="Number of steps to'
fprintf(fid, '                <parameter key="DBCSteps" value="%d" comment="Number of steps to up'
fprintf(fid, '                <parameter key="DtIni" value="%15.12f" comment="Initial time step" /'
fprintf(fid, '                <parameter key="DtMin" value="%15.12f" comment="Minimum time step (c'
fprintf(fid, '                <parameter key="TimeMax" value="0.05" comment="Time of simulation" /'
fprintf(fid, '                <parameter key="TimeOut" value="0.0001" comment="Time between output'
fprintf(fid, '                <parameter key="IncZ" value="0.5" comment="Increase of Z+" />\n');
fprintf(fid, '                <parameter key="PartsOutMax" value="1" comment="Allowed percentage o'
fprintf(fid, '            </parameters>\n');
fprintf(fid, '    </execution>\n');
fprintf(fid, '</case>\n');

message = ferror(fid);

fclose(fid);

system('move Jet.txt Jet.xml'); % Convert to XML

outdir = sprintf('Case-%d',k);
outMake = sprintf('md %s',outdir);
system(outMake);

genRun = sprintf('%s Jet %s/Jet -save:all', gencase,outDir);
dualRun = sprintf('%s %s/Jet %s -svres -gpu', dualsphysics, outDir, outDir);

measure1 = sprintf('%s -dirin %s -filexml %s/Jet.xml -onlytype:-all,+bound -vars:-all,+press
measure2 = sprintf('%s -dirin %s -filexml %s/Jet.xml -onlytype:-all,+fluid -vars:-all,+vel,+

system(genRun);

clc;
fprintf('Running simulation: nr. %d\n',k);

system(dualRun);

system(measure1);
system(measure2);

end

%END OF PROGRAM

```


B.2 Stagnation Pressure Analysis

Pressure analysis of the pressure distribution along the plate and the stagnation pressure.

```
% Post-pro in Matlab

clear all;
clc;

% The pressure distribution caused by a circular jet on the impingement plate
% r is the coordinate in the radial direction which is measured from
% the plate centre outward and d is the jet diameter.

d = 0.2;
dp = 0.002;
rho = 1000;
v = 40;

r = (0:dp:.1);
cc=hsv(12);

figure(1)
axis([0 1 0 0.75])
title('Pressure distribution');
ylabel('Pw/Ps');
xlabel('x/d in the radial direction');
hold on

% Stagnation Deviation
stagErr=zeros(750,1);
stagAvgErr=zeros(750,1);
stagErrLess5=zeros(750,1);
stagErrLess10=zeros(750,1);

for i=1:700

filename1 = sprintf('C:\\kjartafu\\DualSPPhysics-v2.01-windows-64bit\\RUN-DIRECTORY\\Mea

if exist(filename1)
    %do nothing
else
    continue;
end

A=importdata(filename1);
s=size(A);

% % Cp ~
% for ii=5:s(1)
%     plot(A(1,2:s(2)),A(ii,2:s(2))/(0.5*rho*v^2));
%     drawnow
% end

%Cp Avg
```

```

pAvg=zeros(1,s(2)-1);
for j=2:s(2) %Each point
%   if(breakV == 1)
%       break;
%   end
    AvgTemp=0;
    t=0;
    for k=5:s(1)-2 %Each timestep
        if(A(k,j) > 0)
            %Last timestep
            if(B(k,4) ~= 0)
                AvgTemp=AvgTemp+A(k,j);
                t=t+1;
            end
        end
    end
    pAvg(1,j-1)=AvgTemp/t;
end

% if(breakV == 0)
plot(A(1,2:s(2))/d,pAvg(1,:)/(rho*v^2),'color',cc(ceil(rand()*12),:),'LineWidth',2);
drawnow
% end

% % Find stagnation deviation
% stagErrTemp=zeros(85,1);
% for k=15:100 % Time: jet impinging the plate
%     if(size(A,2) == 102)
%         stagErrTemp(k) = abs((A(k,102)-(0.5*rho*v^2))/(0.5*rho*v^2));
%     else
%         stagErrTemp(k) = abs((A(k,101)-(0.5*rho*v^2))/(0.5*rho*v^2));
%     end
%     if(stagErrTemp(k)<0.05) % How many time step inside error of 5%
%         stagErrLess5(i) = stagErrLess5(i)+1;
%     end
%     if(stagErrTemp(k)<0.1) % How many time step inside error of 10%
%         stagErrLess10(i) = stagErrLess10(i)+1;
%     end
%     stagAvgErr(i) = stagAvgErr(i)+stagErrTemp(k);
% end
%
% stagErr(i) = max(stagErrTemp);
% stagAvgErr(i) = stagAvgErr(i)/85; % Average
% stagErrLess5(i) = stagErrLess5(i)/85; % In percent
% stagErrLess10(i) = stagErrLess10(i)/85; % In percent
end

% for i=1:750
%     if((stagAvgErr(i) > 0.01) && (stagAvgErr(i) < 0.5))
%         plot(i,stagAvgErr(i)*100,'o');
%         hold on;
%     end
% end

```

B.3 Stagnation Pressure Plot

A Matlab script create plots from simulation results of a normal water jet impinging a plate.

```
% Post-Processing Tool for Plotting
% =====
clear all;
clc;

%Setup:
L = 0.04;
D = 0.2;
v = 40;
rho = 1e3;

% Load property matrix
P = importdata('C:\\kjartafu\\Matlab\\para.csv');
% Indexation:
% 1=Simulation nr., 2 = C_coef, 3 = dp, 4 = alpha, 5 = shepard, 6 = DBC,
% 7 = dt_min, 8 = h, 9 = C, 10 = Vicosity, 11 = Re, 12 = Ma
% Pre-allocate
cp = zeros(700,150);

for i=1:600

% Import data
filename1 = sprintf('C:\\kjartafu\\DualSPPhysics.v2.01.windows.64bit\\RUN_DIRECTORY\\Mea

if exist(filename1)
    %do nothing
else
    continue;
end

A=importdata(filename1);

%Find cp over time
for j=1:151
    if(size(A,2) == 102)
        cp(i,j) = A(j+3,102)/(rho*v^2);
    else
        cp(i,j) = A(j+3,101)/(rho*v^2);
    end
end

end

%A(4:154,1) = time
figure(1);
for k=1:4
    subplot(2,2,k);
    hold on;
    grid on;
    axis([0 15 0 2.5]);
```

```

        xlabel('tU/L', 'FontSize', 12);
        ylabel('p/\rho U^2', 'FontSize', 12);
    end

    cCoef = [40, 100, 200, 400];
    Ma = [0.5, 0.2, 0.1, 0.05];
    dp = [0.0015, 0.002, 0.003];
    alpha = [8e-6 8e-5 8e-4 8e-3];
    h_p = [0.002250 0.003000 0.004500];
    DBC = [1, 50, 100];
    colorplot = ['r' 'g' 'b' 'k'];

    %Plot over time
    for i=1:700 %KUN EKTE CP
        % COLOR PLOT ETTER EGENSKAP
        % check if good
        if ~(1 && all(cp(i,:) == 0))
            hold on;
            figure(1)
            if(P(i,6) == DBC(1))
                c = colorplot(1);
            elseif(P(i,6) == DBC(2))
                c = colorplot(2);
            elseif(P(i,6) == DBC(3))
                c = colorplot(3);
            end
            if(P(i,2) == cCoef(1))
                subplot(2,2,1);
                plot((A(4:154,1)*v)/L, cp(i,:), c, 'LineWidth', 1.4);
            elseif(P(i,2) == cCoef(2))
                subplot(2,2,2);
                plot((A(4:154,1)*v)/L, cp(i,:), c, 'LineWidth', 1.4);
            elseif(P(i,2) == cCoef(3))
                subplot(2,2,3);
                plot((A(4:154,1)*v)/L, cp(i,:), c, 'LineWidth', 1.4);
            elseif(P(i,2) == cCoef(4))
                subplot(2,2,4);
                plot((A(4:154,1)*v)/L, cp(i,:), c, 'LineWidth', 1.4);
            end
            drawnow;
        end
    end

    figure(1);
    for k=1:4
        subplot(2,2,k);
        title(strcat('Ma = ', {' '}, num2str(Ma(k))), 'FontSize', 14);
        hold on;
        plot([0, 15], [0.5 0.5], '--k', 'LineWidth', 1.7);
        set(gca, 'FontSize', 12)
        %legend('Theory', strcat('C_{coef} = ', {' '}, num2str(cCoef(k))));
    end

```

B.4 Inlet Condition

Results from pitot measurements done by Klemetsen [29] modelled with a number of curved surfaces of cylinders with different initial velocities.

```
clear all;
clc;

%Velocity inlet from jet measurement
%Ref. App. C [Klemetsen 2010]

%=====

r=0.0185;           %Jet radius
size=100;          %Size
dp=r/size;         %Particle spacing
xy=zeros(size+1); %Domain

%Variables from Klemetsen
AA=19.3;
BB=0.0286;
CC=0.042;
DD=0.0369;
EE=-0.0075;
V2=20.55;

%—— Temperate Variable ——
ii=1;
jj=1;
rad=zeros(size+1);

%Find velocity
for i=-r:dp:r      %x-index
    for j=-r:dp:r  %y-index
        dist=sqrt(abs(i)^2+abs(j)^2); %Calculate the distance to center
        rad(ii,jj)=sqrt(i^2+j^2);
        if(dist<=r); %The point is in the circle
            mm=1000*dist;
            %Velocity
            V1=AA+BB*mm+CC*mm^2+DD*mm^3+EE*mm^4;
            V3=V2/(0.0005)*(0.0185-dist);
            if(dist<0.00412)
                xy(ii,jj)=V1;
            else
                if(dist<0.018)
                    xy(ii,jj)=V2;
                else
                    xy(ii,jj)=V3;
                end
            end
        end
        jj=jj+1;
    end
    ii=ii+1;
    jj=1;
end
```

```

        ii=ii+1;
end

% 3D plot
figure(1);
surf(0:2*size,0:2*size,xy);colorbar;

figure(2);
plot(rad,xy);

% Horizontal profile
% Create the geometry

cylPnt=zeros(2,size); % Contains radius and velocity

fprintf('COPY TO XML FILE\n');
fprintf('=====\n\n');
fprintf('dp = %8.6f\n\n',dp);

for i=1:size
    %Insert values
    cylPnt(2,i)=r-i*dp+dp;
    cylPnt(1,i)=xy(size+1,i+1);
end

j=1;
for i=1:size
    %Print coordinates
    if (i<size && (cylPnt(1,size-i+1) ~= cylPnt(1,size-i)))
        fprintf('<setmkfluid mk="%d" />\n', (j-1));
        j=j+1;
    end
    fprintf('<drawcylinder radius="%10.8f" mask="3">\n',cylPnt(2,size-i+1));
    fprintf('<point x="0" y="0" z="0.01" />\n');
    fprintf('<point x="0" y="0" z="0.025" />\n');
    fprintf('</drawcylinder>\n');
end

fprintf('=====\n\n');
fprintf('=====\n\n');

k=1;
for j=1:size
    %Print velocities
    if (j<size && (cylPnt(1,size-j+1) ~= cylPnt(1,size-j)))
        fprintf('<velocity mkfluid="%d" x="0" y="0" z="-%6.4f" />\n', (k-1),cylPnt(1,size-j+1));
        k=k+1;
    end
end

%Plot points
figure(3);
plot(cylPnt(2,:),cylPnt(1,:), 'o');

```

B.5 Pressure at the Wall

Pressure at the centerline of the bucket is analysed, imported data from ParaView.

```
clear all;
clc;

% Import CSV from paraview
% Plot static pressure at the wall

% Load data, one .csv file for each time step
for i=0:200

filename = sprintf('C:\\kjartafu\\DualSPPhysics.v2.01-windows.64bit\\RUN_DIRECTORY\\Klen

if exist(filename)
    %do nothing
else
    continue;
end

A(i+1)=importdata(filename);

end

% Theta range
theta = (round(min(A(1).data(:,3))):round(max(A(1).data(:,3))));
% Preallocating Cp
Cp = zeros(size(A,2),size(theta,2));

%A(:).data(:,2) = CP
%A(:).data(:,3) = Angle

for i=1:size(A,2) % Time steps
    k=zeros(size(theta,2),1);
    for j=1:size(A(i).data,1) % Along curved line
        if(A(i).data(j,2)>0) % Cp
            if(Cp(i,round(A(i).data(j,3))-theta(1)+1) == 0) % Unassigned
                Cp(i,round(A(i).data(j,3))-theta(1)+1) = A(i).data(j,2);
            else % Find the average and insert
                k(round(A(i).data(j,3))-theta(1)+1,1) = k(round(A(i).data(j,3))-theta(1)+1,1) + A(i).data(j,2);
                Cp(i,round(A(i).data(j,3))-theta(1)+1) = Cp(i,round(A(i).data(j,3))-theta(1)+1,1)/k(round(A(i).data(j,3))-theta(1)+1,1);
            end
        end
    end
end
for j=1:size(theta,2)
    if(k(j) > 0)
        Cp(i,j) = Cp(i,j)/k(j);
    end
end
end

CpMax = zeros(1,size(theta,2));
CpAvg = zeros(1,size(theta,2));
```

```

for i=1:size(theta,2) % For all time steps
    CpMax(1,i) = max(Cp(:,i)); % Find max
    k=0;
    for j=1:size(A,2)
        if(Cp(j,i)>0) % Find avarage
            CpAvg(1,i) = CpAvg(1,i) + Cp(j,i);
            k=k+1;
        end
    end
    if(k>0)
        CpAvg(1,i) = CpAvg(1,i)/k;
    end
end

%Plots
%Background color red: 241, green: 235, blue: 239 /255
figure('Color',[1 1 1]);
whitebg([234/255 229/255 236/255]);
axis([0 173 0 .8]);
hold on;
grid on;

% Insert Klemetsens plot
klemetsenTemp = imread('plotKlemetsen.png');
klemetsen = flipdim(klemetsenTemp,1);
imagesc([0 175], [0 .5], klemetsen);

plot(theta,CpAvg,'xk');
% Spline
xx = (theta(1):0.01:theta(size(theta,2)));
yy = spline(theta,CpAvg,xx);
plot(xx,yy,'b');

% Data from experiment
ExpT = [61.5 91 111 126 144 150];
ExpCp = [0.105 0.142 0.210 0.322 0.419 0.404];
plot(ExpT,ExpCp,'or','MarkerSize',8);

% Trendline
poly = polyfit(theta,CpAvg,4);
thetavalues = linspace(theta(1),theta(size(theta,2)),500);
Cpfit = polyval(poly,thetavalues);
% Plot the fitted line
plot(thetavalues,Cpfit,'k','LineWidth',2);

```


B.6 Post-Process of a Plane with Points

A plane with points along the centerline is calculated from MeasureTool. A plane is defined with a pointlist in a text file, and then processed in MeasureTool. An example of a pointlist with 101x101 points:

```
POINTSLIST
-0.08 -0.05 0.0
0.0014 0 -0.001
101 1 101

% Post-Process
% A plane along the centerline, data from MeasureTool

clc;
clear all;

% Water Film Thickness, h

A = importdata('PointsHeight2.Height.asc');

theta = zeros(1, size(A,2)-1);
h = zeros(size(A,1)-3, size(A,2)-1);

% Find relationship between angle and h
for i=2:size(A,2)
    theta(1,i-1) = -1*atan2(A(3,i), A(1,i))*(180/pi);
end

c_r = h./0.0185;

figure(2)
hold on;
plot(theta(1,:), A(4:size(A,1), 2:size(A,2)));

% Pressure

% Load once, if many points
A = importdata('PointPress-Pres.asc');

k=1;
for i=2:size(A,2)
    if((0.06-(sqrt(A(1,i)^2+A(3,i)^2))) < 1e-5)
        press(1,k) = -1*atan2(A(3,i), A(1,i))*(180/pi);
        for j=4:size(A,1)
            press(j-2,k) = A(j,i)/(1e3*20^2);
        end
        k=k+1;
    end
end

figure(1)
axis([30 180 0 20]);
```

```
plot (press (1, :), press (2:size (press, 1), :));
```

Appendix C

Results

Results from stagnation pressure simulations from a horizontal water jet impinging normally upon a vertical plate over time.

Mach number is a dimensionless quantity representing the ratio of flow speed to sound speed, there it is defined as the ratio of the initial speed of the water jet to the artificial speed of sound.

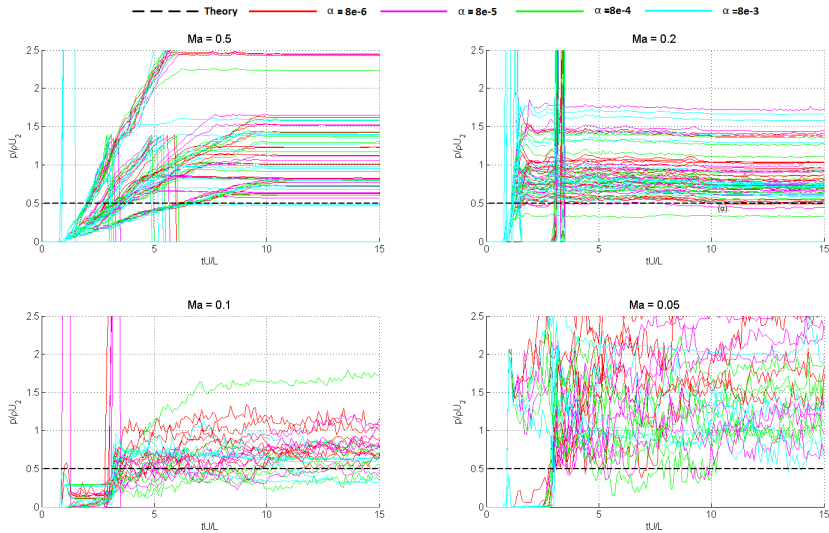


Figure C.0.1: Dimensionless stagnation pressure over time.

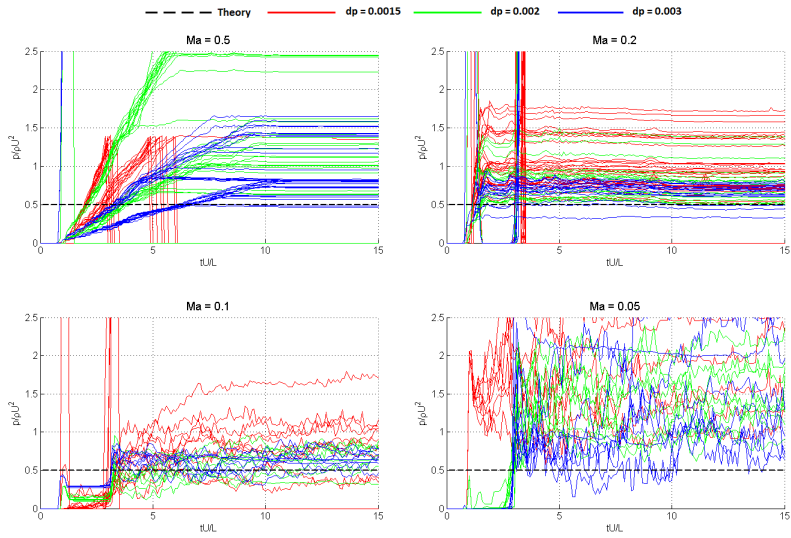


Figure C.0.2: Dimensionless stagnation pressure over time.

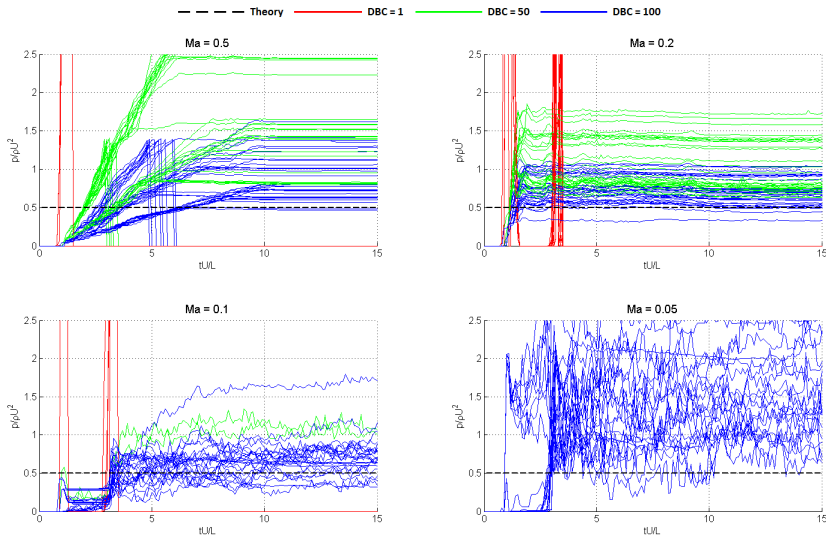


Figure C.0.3: Dimensionless stagnation pressure over time.

Appendix D

Table of Simulation Number

A summary of simulation parameters.

Table D.0.1: Simulation parameters

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
1	40	0.0015	8.0e-006	0	1	5.0e-009
2	40	0.0015	8.0e-006	0	1	1.0e-008
3	40	0.0015	8.0e-006	0	50	5.0e-009
4	40	0.0015	8.0e-006	0	50	1.0e-008
5	40	0.0015	8.0e-006	0	100	5.0e-009
6	40	0.0015	8.0e-006	0	100	1.0e-008
7	40	0.0015	8.0e-006	50	1	5.0e-009
8	40	0.0015	8.0e-006	50	1	1.0e-008
9	40	0.0015	8.0e-006	50	50	5.0e-009
10	40	0.0015	8.0e-006	50	50	1.0e-008
11	40	0.0015	8.0e-006	50	100	5.0e-009
12	40	0.0015	8.0e-006	50	100	1.0e-008
13	40	0.0015	8.0e-005	0	1	5.0e-009
14	40	0.0015	8.0e-005	0	1	1.0e-008
15	40	0.0015	8.0e-005	0	50	5.0e-009
16	40	0.0015	8.0e-005	0	50	1.0e-008
17	40	0.0015	8.0e-005	0	100	5.0e-009
18	40	0.0015	8.0e-005	0	100	1.0e-008
19	40	0.0015	8.0e-005	50	1	5.0e-009
20	40	0.0015	8.0e-005	50	1	1.0e-008
21	40	0.0015	8.0e-005	50	50	5.0e-009
22	40	0.0015	8.0e-005	50	50	1.0e-008
23	40	0.0015	8.0e-005	50	100	5.0e-009
24	40	0.0015	8.0e-005	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
25	40	0.0015	8.0e-004	0	1	5.0e-009
26	40	0.0015	8.0e-004	0	1	1.0e-008
27	40	0.0015	8.0e-004	0	50	5.0e-009
28	40	0.0015	8.0e-004	0	50	1.0e-008
29	40	0.0015	8.0e-004	0	100	5.0e-009
30	40	0.0015	8.0e-004	0	100	1.0e-008
31	40	0.0015	8.0e-004	50	1	5.0e-009
32	40	0.0015	8.0e-004	50	1	1.0e-008
33	40	0.0015	8.0e-004	50	50	5.0e-009
34	40	0.0015	8.0e-004	50	50	1.0e-008
35	40	0.0015	8.0e-004	50	100	5.0e-009
36	40	0.0015	8.0e-004	50	100	1.0e-008
37	40	0.0015	8.0e-003	0	1	5.0e-009
38	40	0.0015	8.0e-003	0	1	1.0e-008
39	40	0.0015	8.0e-003	0	50	5.0e-009
40	40	0.0015	8.0e-003	0	50	1.0e-008
41	40	0.0015	8.0e-003	0	100	5.0e-009
42	40	0.0015	8.0e-003	0	100	1.0e-008
43	40	0.0015	8.0e-003	50	1	5.0e-009
44	40	0.0015	8.0e-003	50	1	1.0e-008
45	40	0.0015	8.0e-003	50	50	5.0e-009
46	40	0.0015	8.0e-003	50	50	1.0e-008
47	40	0.0015	8.0e-003	50	100	5.0e-009
48	40	0.0015	8.0e-003	50	100	1.0e-008
49	40	0.0020	8.0e-006	0	1	5.0e-009
50	40	0.0020	8.0e-006	0	1	1.0e-008
51	40	0.0020	8.0e-006	0	50	5.0e-009
52	40	0.0020	8.0e-006	0	50	1.0e-008
53	40	0.0020	8.0e-006	0	100	5.0e-009
54	40	0.0020	8.0e-006	0	100	1.0e-008
55	40	0.0020	8.0e-006	50	1	5.0e-009
56	40	0.0020	8.0e-006	50	1	1.0e-008
57	40	0.0020	8.0e-006	50	50	5.0e-009
58	40	0.0020	8.0e-006	50	50	1.0e-008
59	40	0.0020	8.0e-006	50	100	5.0e-009
60	40	0.0020	8.0e-006	50	100	1.0e-008
61	40	0.0020	8.0e-005	0	1	5.0e-009
62	40	0.0020	8.0e-005	0	1	1.0e-008
63	40	0.0020	8.0e-005	0	50	5.0e-009
64	40	0.0020	8.0e-005	0	50	1.0e-008
65	40	0.0020	8.0e-005	0	100	5.0e-009
66	40	0.0020	8.0e-005	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
67	40	0.0020	8.0e-005	50	1	5.0e-009
68	40	0.0020	8.0e-005	50	1	1.0e-008
69	40	0.0020	8.0e-005	50	50	5.0e-009
70	40	0.0020	8.0e-005	50	50	1.0e-008
71	40	0.0020	8.0e-005	50	100	5.0e-009
72	40	0.0020	8.0e-005	50	100	1.0e-008
73	40	0.0020	8.0e-004	0	1	5.0e-009
74	40	0.0020	8.0e-004	0	1	1.0e-008
75	40	0.0020	8.0e-004	0	50	5.0e-009
76	40	0.0020	8.0e-004	0	50	1.0e-008
77	40	0.0020	8.0e-004	0	100	5.0e-009
78	40	0.0020	8.0e-004	0	100	1.0e-008
79	40	0.0020	8.0e-004	50	1	5.0e-009
80	40	0.0020	8.0e-004	50	1	1.0e-008
81	40	0.0020	8.0e-004	50	50	5.0e-009
82	40	0.0020	8.0e-004	50	50	1.0e-008
83	40	0.0020	8.0e-004	50	100	5.0e-009
84	40	0.0020	8.0e-004	50	100	1.0e-008
85	40	0.0020	8.0e-003	0	1	5.0e-009
86	40	0.0020	8.0e-003	0	1	1.0e-008
87	40	0.0020	8.0e-003	0	50	5.0e-009
88	40	0.0020	8.0e-003	0	50	1.0e-008
89	40	0.0020	8.0e-003	0	100	5.0e-009
90	40	0.0020	8.0e-003	0	100	1.0e-008
91	40	0.0020	8.0e-003	50	1	5.0e-009
92	40	0.0020	8.0e-003	50	1	1.0e-008
93	40	0.0020	8.0e-003	50	50	5.0e-009
94	40	0.0020	8.0e-003	50	50	1.0e-008
95	40	0.0020	8.0e-003	50	100	5.0e-009
96	40	0.0020	8.0e-003	50	100	1.0e-008
97	40	0.0030	8.0e-006	0	1	5.0e-009
98	40	0.0030	8.0e-006	0	1	1.0e-008
99	40	0.0030	8.0e-006	0	50	5.0e-009
100	40	0.0030	8.0e-006	0	50	1.0e-008
101	40	0.0030	8.0e-006	0	100	5.0e-009
102	40	0.0030	8.0e-006	0	100	1.0e-008
103	40	0.0030	8.0e-006	50	1	5.0e-009
104	40	0.0030	8.0e-006	50	1	1.0e-008
105	40	0.0030	8.0e-006	50	50	5.0e-009
106	40	0.0030	8.0e-006	50	50	1.0e-008
107	40	0.0030	8.0e-006	50	100	5.0e-009
108	40	0.0030	8.0e-006	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
109	40	0.0030	8.0e-005	0	1	5.0e-009
110	40	0.0030	8.0e-005	0	1	1.0e-008
111	40	0.0030	8.0e-005	0	50	5.0e-009
112	40	0.0030	8.0e-005	0	50	1.0e-008
113	40	0.0030	8.0e-005	0	100	5.0e-009
114	40	0.0030	8.0e-005	0	100	1.0e-008
115	40	0.0030	8.0e-005	50	1	5.0e-009
116	40	0.0030	8.0e-005	50	1	1.0e-008
117	40	0.0030	8.0e-005	50	50	5.0e-009
118	40	0.0030	8.0e-005	50	50	1.0e-008
119	40	0.0030	8.0e-005	50	100	5.0e-009
120	40	0.0030	8.0e-005	50	100	1.0e-008
121	40	0.0030	8.0e-004	0	1	5.0e-009
122	40	0.0030	8.0e-004	0	1	1.0e-008
123	40	0.0030	8.0e-004	0	50	5.0e-009
124	40	0.0030	8.0e-004	0	50	1.0e-008
125	40	0.0030	8.0e-004	0	100	5.0e-009
126	40	0.0030	8.0e-004	0	100	1.0e-008
127	40	0.0030	8.0e-004	50	1	5.0e-009
128	40	0.0030	8.0e-004	50	1	1.0e-008
129	40	0.0030	8.0e-004	50	50	5.0e-009
130	40	0.0030	8.0e-004	50	50	1.0e-008
131	40	0.0030	8.0e-004	50	100	5.0e-009
132	40	0.0030	8.0e-004	50	100	1.0e-008
133	40	0.0030	8.0e-003	0	1	5.0e-009
134	40	0.0030	8.0e-003	0	1	1.0e-008
135	40	0.0030	8.0e-003	0	50	5.0e-009
136	40	0.0030	8.0e-003	0	50	1.0e-008
137	40	0.0030	8.0e-003	0	100	5.0e-009
138	40	0.0030	8.0e-003	0	100	1.0e-008
139	40	0.0030	8.0e-003	50	1	5.0e-009
140	40	0.0030	8.0e-003	50	1	1.0e-008
141	40	0.0030	8.0e-003	50	50	5.0e-009
142	40	0.0030	8.0e-003	50	50	1.0e-008
143	40	0.0030	8.0e-003	50	100	5.0e-009
144	40	0.0030	8.0e-003	50	100	1.0e-008
145	100	0.0015	8.0e-006	0	1	5.0e-009
146	100	0.0015	8.0e-006	0	1	1.0e-008
147	100	0.0015	8.0e-006	0	50	5.0e-009
148	100	0.0015	8.0e-006	0	50	1.0e-008
149	100	0.0015	8.0e-006	0	100	5.0e-009
150	100	0.0015	8.0e-006	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
151	100	0.0015	8.0e-006	50	1	5.0e-009
152	100	0.0015	8.0e-006	50	1	1.0e-008
153	100	0.0015	8.0e-006	50	50	5.0e-009
154	100	0.0015	8.0e-006	50	50	1.0e-008
155	100	0.0015	8.0e-006	50	100	5.0e-009
156	100	0.0015	8.0e-006	50	100	1.0e-008
157	100	0.0015	8.0e-005	0	1	5.0e-009
158	100	0.0015	8.0e-005	0	1	1.0e-008
159	100	0.0015	8.0e-005	0	50	5.0e-009
160	100	0.0015	8.0e-005	0	50	1.0e-008
161	100	0.0015	8.0e-005	0	100	5.0e-009
162	100	0.0015	8.0e-005	0	100	1.0e-008
163	100	0.0015	8.0e-005	50	1	5.0e-009
164	100	0.0015	8.0e-005	50	1	1.0e-008
165	100	0.0015	8.0e-005	50	50	5.0e-009
166	100	0.0015	8.0e-005	50	50	1.0e-008
167	100	0.0015	8.0e-005	50	100	5.0e-009
168	100	0.0015	8.0e-005	50	100	1.0e-008
169	100	0.0015	8.0e-004	0	1	5.0e-009
170	100	0.0015	8.0e-004	0	1	1.0e-008
171	100	0.0015	8.0e-004	0	50	5.0e-009
172	100	0.0015	8.0e-004	0	50	1.0e-008
173	100	0.0015	8.0e-004	0	100	5.0e-009
174	100	0.0015	8.0e-004	0	100	1.0e-008
175	100	0.0015	8.0e-004	50	1	5.0e-009
176	100	0.0015	8.0e-004	50	1	1.0e-008
177	100	0.0015	8.0e-004	50	50	5.0e-009
178	100	0.0015	8.0e-004	50	50	1.0e-008
179	100	0.0015	8.0e-004	50	100	5.0e-009
180	100	0.0015	8.0e-004	50	100	1.0e-008
181	100	0.0015	8.0e-003	0	1	5.0e-009
182	100	0.0015	8.0e-003	0	1	1.0e-008
183	100	0.0015	8.0e-003	0	50	5.0e-009
184	100	0.0015	8.0e-003	0	50	1.0e-008
185	100	0.0015	8.0e-003	0	100	5.0e-009
186	100	0.0015	8.0e-003	0	100	1.0e-008
187	100	0.0015	8.0e-003	50	1	5.0e-009
188	100	0.0015	8.0e-003	50	1	1.0e-008
189	100	0.0015	8.0e-003	50	50	5.0e-009
190	100	0.0015	8.0e-003	50	50	1.0e-008
191	100	0.0015	8.0e-003	50	100	5.0e-009
192	100	0.0015	8.0e-003	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
193	100	0.0020	8.0e-006	0	1	5.0e-009
194	100	0.0020	8.0e-006	0	1	1.0e-008
195	100	0.0020	8.0e-006	0	50	5.0e-009
196	100	0.0020	8.0e-006	0	50	1.0e-008
197	100	0.0020	8.0e-006	0	100	5.0e-009
198	100	0.0020	8.0e-006	0	100	1.0e-008
199	100	0.0020	8.0e-006	50	1	5.0e-009
200	100	0.0020	8.0e-006	50	1	1.0e-008
201	100	0.0020	8.0e-006	50	50	5.0e-009
202	100	0.0020	8.0e-006	50	50	1.0e-008
203	100	0.0020	8.0e-006	50	100	5.0e-009
204	100	0.0020	8.0e-006	50	100	1.0e-008
205	100	0.0020	8.0e-005	0	1	5.0e-009
206	100	0.0020	8.0e-005	0	1	1.0e-008
207	100	0.0020	8.0e-005	0	50	5.0e-009
208	100	0.0020	8.0e-005	0	50	1.0e-008
209	100	0.0020	8.0e-005	0	100	5.0e-009
210	100	0.0020	8.0e-005	0	100	1.0e-008
211	100	0.0020	8.0e-005	50	1	5.0e-009
212	100	0.0020	8.0e-005	50	1	1.0e-008
213	100	0.0020	8.0e-005	50	50	5.0e-009
214	100	0.0020	8.0e-005	50	50	1.0e-008
215	100	0.0020	8.0e-005	50	100	5.0e-009
216	100	0.0020	8.0e-005	50	100	1.0e-008
217	100	0.0020	8.0e-004	0	1	5.0e-009
218	100	0.0020	8.0e-004	0	1	1.0e-008
219	100	0.0020	8.0e-004	0	50	5.0e-009
220	100	0.0020	8.0e-004	0	50	1.0e-008
221	100	0.0020	8.0e-004	0	100	5.0e-009
222	100	0.0020	8.0e-004	0	100	1.0e-008
223	100	0.0020	8.0e-004	50	1	5.0e-009
224	100	0.0020	8.0e-004	50	1	1.0e-008
225	100	0.0020	8.0e-004	50	50	5.0e-009
226	100	0.0020	8.0e-004	50	50	1.0e-008
227	100	0.0020	8.0e-004	50	100	5.0e-009
228	100	0.0020	8.0e-004	50	100	1.0e-008
229	100	0.0020	8.0e-003	0	1	5.0e-009
230	100	0.0020	8.0e-003	0	1	1.0e-008
231	100	0.0020	8.0e-003	0	50	5.0e-009
232	100	0.0020	8.0e-003	0	50	1.0e-008
233	100	0.0020	8.0e-003	0	100	5.0e-009
234	100	0.0020	8.0e-003	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
235	100	0.0020	8.0e-003	50	1	5.0e-009
236	100	0.0020	8.0e-003	50	1	1.0e-008
237	100	0.0020	8.0e-003	50	50	5.0e-009
238	100	0.0020	8.0e-003	50	50	1.0e-008
239	100	0.0020	8.0e-003	50	100	5.0e-009
240	100	0.0020	8.0e-003	50	100	1.0e-008
241	100	0.0030	8.0e-006	0	1	5.0e-009
242	100	0.0030	8.0e-006	0	1	1.0e-008
243	100	0.0030	8.0e-006	0	50	5.0e-009
244	100	0.0030	8.0e-006	0	50	1.0e-008
245	100	0.0030	8.0e-006	0	100	5.0e-009
246	100	0.0030	8.0e-006	0	100	1.0e-008
247	100	0.0030	8.0e-006	50	1	5.0e-009
248	100	0.0030	8.0e-006	50	1	1.0e-008
249	100	0.0030	8.0e-006	50	50	5.0e-009
250	100	0.0030	8.0e-006	50	50	1.0e-008
251	100	0.0030	8.0e-006	50	100	5.0e-009
252	100	0.0030	8.0e-006	50	100	1.0e-008
253	100	0.0030	8.0e-005	0	1	5.0e-009
254	100	0.0030	8.0e-005	0	1	1.0e-008
255	100	0.0030	8.0e-005	0	50	5.0e-009
256	100	0.0030	8.0e-005	0	50	1.0e-008
257	100	0.0030	8.0e-005	0	100	5.0e-009
258	100	0.0030	8.0e-005	0	100	1.0e-008
259	100	0.0030	8.0e-005	50	1	5.0e-009
260	100	0.0030	8.0e-005	50	1	1.0e-008
261	100	0.0030	8.0e-005	50	50	5.0e-009
262	100	0.0030	8.0e-005	50	50	1.0e-008
263	100	0.0030	8.0e-005	50	100	5.0e-009
264	100	0.0030	8.0e-005	50	100	1.0e-008
265	100	0.0030	8.0e-004	0	1	5.0e-009
266	100	0.0030	8.0e-004	0	1	1.0e-008
267	100	0.0030	8.0e-004	0	50	5.0e-009
268	100	0.0030	8.0e-004	0	50	1.0e-008
269	100	0.0030	8.0e-004	0	100	5.0e-009
270	100	0.0030	8.0e-004	0	100	1.0e-008
271	100	0.0030	8.0e-004	50	1	5.0e-009
272	100	0.0030	8.0e-004	50	1	1.0e-008
273	100	0.0030	8.0e-004	50	50	5.0e-009
274	100	0.0030	8.0e-004	50	50	1.0e-008
275	100	0.0030	8.0e-004	50	100	5.0e-009
276	100	0.0030	8.0e-004	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
277	100	0.0030	8.0e-003	0	1	5.0e-009
278	100	0.0030	8.0e-003	0	1	1.0e-008
279	100	0.0030	8.0e-003	0	50	5.0e-009
280	100	0.0030	8.0e-003	0	50	1.0e-008
281	100	0.0030	8.0e-003	0	100	5.0e-009
282	100	0.0030	8.0e-003	0	100	1.0e-008
283	100	0.0030	8.0e-003	50	1	5.0e-009
284	100	0.0030	8.0e-003	50	1	1.0e-008
285	100	0.0030	8.0e-003	50	50	5.0e-009
286	100	0.0030	8.0e-003	50	50	1.0e-008
287	100	0.0030	8.0e-003	50	100	5.0e-009
288	100	0.0030	8.0e-003	50	100	1.0e-008
289	200	0.0015	8.0e-006	0	1	5.0e-009
290	200	0.0015	8.0e-006	0	1	1.0e-008
291	200	0.0015	8.0e-006	0	50	5.0e-009
292	200	0.0015	8.0e-006	0	50	1.0e-008
293	200	0.0015	8.0e-006	0	100	5.0e-009
294	200	0.0015	8.0e-006	0	100	1.0e-008
295	200	0.0015	8.0e-006	50	1	5.0e-009
296	200	0.0015	8.0e-006	50	1	1.0e-008
297	200	0.0015	8.0e-006	50	50	5.0e-009
298	200	0.0015	8.0e-006	50	50	1.0e-008
299	200	0.0015	8.0e-006	50	100	5.0e-009
300	200	0.0015	8.0e-006	50	100	1.0e-008
301	200	0.0015	8.0e-005	0	1	5.0e-009
302	200	0.0015	8.0e-005	0	1	1.0e-008
303	200	0.0015	8.0e-005	0	50	5.0e-009
304	200	0.0015	8.0e-005	0	50	1.0e-008
305	200	0.0015	8.0e-005	0	100	5.0e-009
306	200	0.0015	8.0e-005	0	100	1.0e-008
307	200	0.0015	8.0e-005	50	1	5.0e-009
308	200	0.0015	8.0e-005	50	1	1.0e-008
309	200	0.0015	8.0e-005	50	50	5.0e-009
310	200	0.0015	8.0e-005	50	50	1.0e-008
311	200	0.0015	8.0e-005	50	100	5.0e-009
312	200	0.0015	8.0e-005	50	100	1.0e-008
313	200	0.0015	8.0e-004	0	1	5.0e-009
314	200	0.0015	8.0e-004	0	1	1.0e-008
315	200	0.0015	8.0e-004	0	50	5.0e-009
316	200	0.0015	8.0e-004	0	50	1.0e-008
317	200	0.0015	8.0e-004	0	100	5.0e-009
318	200	0.0015	8.0e-004	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
319	200	0.0015	8.0e-004	50	1	5.0e-009
320	200	0.0015	8.0e-004	50	1	1.0e-008
321	200	0.0015	8.0e-004	50	50	5.0e-009
322	200	0.0015	8.0e-004	50	50	1.0e-008
323	200	0.0015	8.0e-004	50	100	5.0e-009
324	200	0.0015	8.0e-004	50	100	1.0e-008
325	200	0.0015	8.0e-003	0	1	5.0e-009
326	200	0.0015	8.0e-003	0	1	1.0e-008
327	200	0.0015	8.0e-003	0	50	5.0e-009
328	200	0.0015	8.0e-003	0	50	1.0e-008
329	200	0.0015	8.0e-003	0	100	5.0e-009
330	200	0.0015	8.0e-003	0	100	1.0e-008
331	200	0.0015	8.0e-003	50	1	5.0e-009
332	200	0.0015	8.0e-003	50	1	1.0e-008
333	200	0.0015	8.0e-003	50	50	5.0e-009
334	200	0.0015	8.0e-003	50	50	1.0e-008
335	200	0.0015	8.0e-003	50	100	5.0e-009
336	200	0.0015	8.0e-003	50	100	1.0e-008
337	200	0.0020	8.0e-006	0	1	5.0e-009
338	200	0.0020	8.0e-006	0	1	1.0e-008
339	200	0.0020	8.0e-006	0	50	5.0e-009
340	200	0.0020	8.0e-006	0	50	1.0e-008
341	200	0.0020	8.0e-006	0	100	5.0e-009
342	200	0.0020	8.0e-006	0	100	1.0e-008
343	200	0.0020	8.0e-006	50	1	5.0e-009
344	200	0.0020	8.0e-006	50	1	1.0e-008
345	200	0.0020	8.0e-006	50	50	5.0e-009
346	200	0.0020	8.0e-006	50	50	1.0e-008
347	200	0.0020	8.0e-006	50	100	5.0e-009
348	200	0.0020	8.0e-006	50	100	1.0e-008
349	200	0.0020	8.0e-005	0	1	5.0e-009
350	200	0.0020	8.0e-005	0	1	1.0e-008
351	200	0.0020	8.0e-005	0	50	5.0e-009
352	200	0.0020	8.0e-005	0	50	1.0e-008
353	200	0.0020	8.0e-005	0	100	5.0e-009
354	200	0.0020	8.0e-005	0	100	1.0e-008
355	200	0.0020	8.0e-005	50	1	5.0e-009
356	200	0.0020	8.0e-005	50	1	1.0e-008
357	200	0.0020	8.0e-005	50	50	5.0e-009
358	200	0.0020	8.0e-005	50	50	1.0e-008
359	200	0.0020	8.0e-005	50	100	5.0e-009
360	200	0.0020	8.0e-005	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
361	200	0.0020	8.0e-004	0	1	5.0e-009
362	200	0.0020	8.0e-004	0	1	1.0e-008
363	200	0.0020	8.0e-004	0	50	5.0e-009
364	200	0.0020	8.0e-004	0	50	1.0e-008
365	200	0.0020	8.0e-004	0	100	5.0e-009
366	200	0.0020	8.0e-004	0	100	1.0e-008
367	200	0.0020	8.0e-004	50	1	5.0e-009
368	200	0.0020	8.0e-004	50	1	1.0e-008
369	200	0.0020	8.0e-004	50	50	5.0e-009
370	200	0.0020	8.0e-004	50	50	1.0e-008
371	200	0.0020	8.0e-004	50	100	5.0e-009
372	200	0.0020	8.0e-004	50	100	1.0e-008
373	200	0.0020	8.0e-003	0	1	5.0e-009
374	200	0.0020	8.0e-003	0	1	1.0e-008
375	200	0.0020	8.0e-003	0	50	5.0e-009
376	200	0.0020	8.0e-003	0	50	1.0e-008
377	200	0.0020	8.0e-003	0	100	5.0e-009
378	200	0.0020	8.0e-003	0	100	1.0e-008
379	200	0.0020	8.0e-003	50	1	5.0e-009
380	200	0.0020	8.0e-003	50	1	1.0e-008
381	200	0.0020	8.0e-003	50	50	5.0e-009
382	200	0.0020	8.0e-003	50	50	1.0e-008
383	200	0.0020	8.0e-003	50	100	5.0e-009
384	200	0.0020	8.0e-003	50	100	1.0e-008
385	200	0.0030	8.0e-006	0	1	5.0e-009
386	200	0.0030	8.0e-006	0	1	1.0e-008
387	200	0.0030	8.0e-006	0	50	5.0e-009
388	200	0.0030	8.0e-006	0	50	1.0e-008
389	200	0.0030	8.0e-006	0	100	5.0e-009
390	200	0.0030	8.0e-006	0	100	1.0e-008
391	200	0.0030	8.0e-006	50	1	5.0e-009
392	200	0.0030	8.0e-006	50	1	1.0e-008
393	200	0.0030	8.0e-006	50	50	5.0e-009
394	200	0.0030	8.0e-006	50	50	1.0e-008
395	200	0.0030	8.0e-006	50	100	5.0e-009
396	200	0.0030	8.0e-006	50	100	1.0e-008
397	200	0.0030	8.0e-005	0	1	5.0e-009
398	200	0.0030	8.0e-005	0	1	1.0e-008
399	200	0.0030	8.0e-005	0	50	5.0e-009
400	200	0.0030	8.0e-005	0	50	1.0e-008
401	200	0.0030	8.0e-005	0	100	5.0e-009
402	200	0.0030	8.0e-005	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
403	200	0.0030	8.0e-005	50	1	5.0e-009
404	200	0.0030	8.0e-005	50	1	1.0e-008
405	200	0.0030	8.0e-005	50	50	5.0e-009
406	200	0.0030	8.0e-005	50	50	1.0e-008
407	200	0.0030	8.0e-005	50	100	5.0e-009
408	200	0.0030	8.0e-005	50	100	1.0e-008
409	200	0.0030	8.0e-004	0	1	5.0e-009
410	200	0.0030	8.0e-004	0	1	1.0e-008
411	200	0.0030	8.0e-004	0	50	5.0e-009
412	200	0.0030	8.0e-004	0	50	1.0e-008
413	200	0.0030	8.0e-004	0	100	5.0e-009
414	200	0.0030	8.0e-004	0	100	1.0e-008
415	200	0.0030	8.0e-004	50	1	5.0e-009
416	200	0.0030	8.0e-004	50	1	1.0e-008
417	200	0.0030	8.0e-004	50	50	5.0e-009
418	200	0.0030	8.0e-004	50	50	1.0e-008
419	200	0.0030	8.0e-004	50	100	5.0e-009
420	200	0.0030	8.0e-004	50	100	1.0e-008
421	200	0.0030	8.0e-003	0	1	5.0e-009
422	200	0.0030	8.0e-003	0	1	1.0e-008
423	200	0.0030	8.0e-003	0	50	5.0e-009
424	200	0.0030	8.0e-003	0	50	1.0e-008
425	200	0.0030	8.0e-003	0	100	5.0e-009
426	200	0.0030	8.0e-003	0	100	1.0e-008
427	200	0.0030	8.0e-003	50	1	5.0e-009
428	200	0.0030	8.0e-003	50	1	1.0e-008
429	200	0.0030	8.0e-003	50	50	5.0e-009
430	200	0.0030	8.0e-003	50	50	1.0e-008
431	200	0.0030	8.0e-003	50	100	5.0e-009
432	200	0.0030	8.0e-003	50	100	1.0e-008
433	400	0.0015	8.0e-006	0	1	5.0e-009
434	400	0.0015	8.0e-006	0	1	1.0e-008
435	400	0.0015	8.0e-006	0	50	5.0e-009
436	400	0.0015	8.0e-006	0	50	1.0e-008
437	400	0.0015	8.0e-006	0	100	5.0e-009
438	400	0.0015	8.0e-006	0	100	1.0e-008
439	400	0.0015	8.0e-006	50	1	5.0e-009
440	400	0.0015	8.0e-006	50	1	1.0e-008
441	400	0.0015	8.0e-006	50	50	5.0e-009
442	400	0.0015	8.0e-006	50	50	1.0e-008
443	400	0.0015	8.0e-006	50	100	5.0e-009
444	400	0.0015	8.0e-006	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
445	400	0.0015	8.0e-005	0	1	5.0e-009
446	400	0.0015	8.0e-005	0	1	1.0e-008
447	400	0.0015	8.0e-005	0	50	5.0e-009
448	400	0.0015	8.0e-005	0	50	1.0e-008
449	400	0.0015	8.0e-005	0	100	5.0e-009
450	400	0.0015	8.0e-005	0	100	1.0e-008
451	400	0.0015	8.0e-005	50	1	5.0e-009
452	400	0.0015	8.0e-005	50	1	1.0e-008
453	400	0.0015	8.0e-005	50	50	5.0e-009
454	400	0.0015	8.0e-005	50	50	1.0e-008
455	400	0.0015	8.0e-005	50	100	5.0e-009
456	400	0.0015	8.0e-005	50	100	1.0e-008
457	400	0.0015	8.0e-004	0	1	5.0e-009
458	400	0.0015	8.0e-004	0	1	1.0e-008
459	400	0.0015	8.0e-004	0	50	5.0e-009
460	400	0.0015	8.0e-004	0	50	1.0e-008
461	400	0.0015	8.0e-004	0	100	5.0e-009
462	400	0.0015	8.0e-004	0	100	1.0e-008
463	400	0.0015	8.0e-004	50	1	5.0e-009
464	400	0.0015	8.0e-004	50	1	1.0e-008
465	400	0.0015	8.0e-004	50	50	5.0e-009
466	400	0.0015	8.0e-004	50	50	1.0e-008
467	400	0.0015	8.0e-004	50	100	5.0e-009
468	400	0.0015	8.0e-004	50	100	1.0e-008
469	400	0.0015	8.0e-003	0	1	5.0e-009
470	400	0.0015	8.0e-003	0	1	1.0e-008
471	400	0.0015	8.0e-003	0	50	5.0e-009
472	400	0.0015	8.0e-003	0	50	1.0e-008
473	400	0.0015	8.0e-003	0	100	5.0e-009
474	400	0.0015	8.0e-003	0	100	1.0e-008
475	400	0.0015	8.0e-003	50	1	5.0e-009
476	400	0.0015	8.0e-003	50	1	1.0e-008
477	400	0.0015	8.0e-003	50	50	5.0e-009
478	400	0.0015	8.0e-003	50	50	1.0e-008
479	400	0.0015	8.0e-003	50	100	5.0e-009
480	400	0.0015	8.0e-003	50	100	1.0e-008
481	400	0.0020	8.0e-006	0	1	5.0e-009
482	400	0.0020	8.0e-006	0	1	1.0e-008
483	400	0.0020	8.0e-006	0	50	5.0e-009
484	400	0.0020	8.0e-006	0	50	1.0e-008
485	400	0.0020	8.0e-006	0	100	5.0e-009
486	400	0.0020	8.0e-006	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
487	400	0.0020	8.0e-006	50	1	5.0e-009
488	400	0.0020	8.0e-006	50	1	1.0e-008
489	400	0.0020	8.0e-006	50	50	5.0e-009
490	400	0.0020	8.0e-006	50	50	1.0e-008
491	400	0.0020	8.0e-006	50	100	5.0e-009
492	400	0.0020	8.0e-006	50	100	1.0e-008
493	400	0.0020	8.0e-005	0	1	5.0e-009
494	400	0.0020	8.0e-005	0	1	1.0e-008
495	400	0.0020	8.0e-005	0	50	5.0e-009
496	400	0.0020	8.0e-005	0	50	1.0e-008
497	400	0.0020	8.0e-005	0	100	5.0e-009
498	400	0.0020	8.0e-005	0	100	1.0e-008
499	400	0.0020	8.0e-005	50	1	5.0e-009
500	400	0.0020	8.0e-005	50	1	1.0e-008
501	400	0.0020	8.0e-005	50	50	5.0e-009
502	400	0.0020	8.0e-005	50	50	1.0e-008
503	400	0.0020	8.0e-005	50	100	5.0e-009
504	400	0.0020	8.0e-005	50	100	1.0e-008
505	400	0.0020	8.0e-004	0	1	5.0e-009
506	400	0.0020	8.0e-004	0	1	1.0e-008
507	400	0.0020	8.0e-004	0	50	5.0e-009
508	400	0.0020	8.0e-004	0	50	1.0e-008
509	400	0.0020	8.0e-004	0	100	5.0e-009
510	400	0.0020	8.0e-004	0	100	1.0e-008
511	400	0.0020	8.0e-004	50	1	5.0e-009
512	400	0.0020	8.0e-004	50	1	1.0e-008
513	400	0.0020	8.0e-004	50	50	5.0e-009
514	400	0.0020	8.0e-004	50	50	1.0e-008
515	400	0.0020	8.0e-004	50	100	5.0e-009
516	400	0.0020	8.0e-004	50	100	1.0e-008
517	400	0.0020	8.0e-003	0	1	5.0e-009
518	400	0.0020	8.0e-003	0	1	1.0e-008
519	400	0.0020	8.0e-003	0	50	5.0e-009
520	400	0.0020	8.0e-003	0	50	1.0e-008
521	400	0.0020	8.0e-003	0	100	5.0e-009
522	400	0.0020	8.0e-003	0	100	1.0e-008
523	400	0.0020	8.0e-003	50	1	5.0e-009
524	400	0.0020	8.0e-003	50	1	1.0e-008
525	400	0.0020	8.0e-003	50	50	5.0e-009
526	400	0.0020	8.0e-003	50	50	1.0e-008
527	400	0.0020	8.0e-003	50	100	5.0e-009
528	400	0.0020	8.0e-003	50	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
529	400	0.0030	8.0e-006	0	1	5.0e-009
530	400	0.0030	8.0e-006	0	1	1.0e-008
531	400	0.0030	8.0e-006	0	50	5.0e-009
532	400	0.0030	8.0e-006	0	50	1.0e-008
533	400	0.0030	8.0e-006	0	100	5.0e-009
534	400	0.0030	8.0e-006	0	100	1.0e-008
535	400	0.0030	8.0e-006	50	1	5.0e-009
536	400	0.0030	8.0e-006	50	1	1.0e-008
537	400	0.0030	8.0e-006	50	50	5.0e-009
538	400	0.0030	8.0e-006	50	50	1.0e-008
539	400	0.0030	8.0e-006	50	100	5.0e-009
540	400	0.0030	8.0e-006	50	100	1.0e-008
541	400	0.0030	8.0e-005	0	1	5.0e-009
542	400	0.0030	8.0e-005	0	1	1.0e-008
543	400	0.0030	8.0e-005	0	50	5.0e-009
544	400	0.0030	8.0e-005	0	50	1.0e-008
545	400	0.0030	8.0e-005	0	100	5.0e-009
546	400	0.0030	8.0e-005	0	100	1.0e-008
547	400	0.0030	8.0e-005	50	1	5.0e-009
548	400	0.0030	8.0e-005	50	1	1.0e-008
549	400	0.0030	8.0e-005	50	50	5.0e-009
550	400	0.0030	8.0e-005	50	50	1.0e-008
551	400	0.0030	8.0e-005	50	100	5.0e-009
552	400	0.0030	8.0e-005	50	100	1.0e-008
553	400	0.0030	8.0e-004	0	1	5.0e-009
554	400	0.0030	8.0e-004	0	1	1.0e-008
555	400	0.0030	8.0e-004	0	50	5.0e-009
556	400	0.0030	8.0e-004	0	50	1.0e-008
557	400	0.0030	8.0e-004	0	100	5.0e-009
558	400	0.0030	8.0e-004	0	100	1.0e-008
559	400	0.0030	8.0e-004	50	1	5.0e-009
560	400	0.0030	8.0e-004	50	1	1.0e-008
561	400	0.0030	8.0e-004	50	50	5.0e-009
562	400	0.0030	8.0e-004	50	50	1.0e-008
563	400	0.0030	8.0e-004	50	100	5.0e-009
564	400	0.0030	8.0e-004	50	100	1.0e-008
565	400	0.0030	8.0e-003	0	1	5.0e-009
566	400	0.0030	8.0e-003	0	1	1.0e-008
567	400	0.0030	8.0e-003	0	50	5.0e-009
568	400	0.0030	8.0e-003	0	50	1.0e-008
569	400	0.0030	8.0e-003	0	100	5.0e-009
570	400	0.0030	8.0e-003	0	100	1.0e-008

Continued on next page

Table D.0.1 – *Continued from previous page*

Sim.nr	C_{coef}	dp	α	Shepard	DBC	dt_{min}
571	400	0.0030	8.0e-003	50	1	5.0e-009
572	400	0.0030	8.0e-003	50	1	1.0e-008
573	400	0.0030	8.0e-003	50	50	5.0e-009
574	400	0.0030	8.0e-003	50	50	1.0e-008
575	400	0.0030	8.0e-003	50	100	5.0e-009
576	400	0.0030	8.0e-003	50	100	1.0e-008