Finite Particle Flow Simulation of Free Jet Deviation by Rotating Pelton Buckets

THÈSE N^{O} 6470 (2015)

PRÉSENTÉE LE 30 JANVIER 2015
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE MACHINES HYDRAULIQUES
PROGRAMME DOCTORAL EN ENERGIE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Christian VESSAZ

acceptée sur proposition du jury:

Prof. M. Paolone, président du jury
Prof. F. Avellan, Prof. C. Tournier, directeurs de thèse
Prof. F. De Vuyst, rapporteur
Prof. M. Picasso, rapporteur
Dr C. Segoufin, rapporteuse

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Our greatest weakness lies in giving up.
The most certain way to succeed is always to try just one more time.
— Thomas A. Edison

Acknowledgements

I would like to thank the members of the jury, Prof. Mario Paolone, Prof. François Avellan, Prof. Christophe Tournier, Prof. Florian De Vuyst, Prof. Marco Picasso and Dr Claire Segoufin for their valuable time and constructive discussions during the private defense.

The research leading to the results published in this thesis has received funding from the Ark, the foundation for innovation of Valais Canton, through the project HydroVS. I would like also to acknowledge the financial support and technical assistance of ALSTOM Power Hydro for the development of the *SPHEROS* software.

The Center for ADvanced MOdelling Science (CADMOS) provided the opportunity to perform numerical simulations on the *Lemanicus* BG/Q at EPFL. The financial support for CADMOS and the Blue Gene/Q system is provided by the Canton of Geneva, Canton of Vaud, Hans Wilsdorf Foundation, Louis-Jeantet Foundation, University of Geneva, University of Lausanne, and École polytechnique fédérale de Lausanne.

Je me permets de poursuivre mes remerciements de manière moins formel en français. Tout d'abord, je tiens à remercier mon directeur de thèse, le Prof. François Avellan, de m'avoir proposé d'effectuer un travail de doctorat dans son laboratoire. Je le remercie sincèrement de m'avoir guidé dans ma recherche et de m'avoir donné la chance de présenter mes travaux lors de multiples congrès internationaux.

Je remercie également mon co-directeur de thèse, le Prof. Christophe Tournier, pour les discussions intéressantes sur la création de géométrie et l'usinage de ces dernières. Je le remercie tout particulièrement pour son regard externe sur les méthodes particulaires et ses précieux commentaires.

Un grand merci à Isabelle Stoudmann-Schmutz pour le support administratif, ainsi qu'à Philippe Cerrutti pour le support informatique sur les machines de calculs du LMH. Je remercie également tous les membres du bureau d'études, les membres du groupe essais machines, ainsi que les mécaniciens du LMH pour tous les moments de convivialité passés ensemble que ce soit à la cafétéria ou lors des soupers du labo. En particulier, je remercie Maxime Raton, Victor Rivas, David Buzzi, Vincent Berruex, Sébastien Bourgeois et Alberto Bullani pour les matchs de beach-volley, les grillades et les discussions animées au Great Escape.

Je remercie les Dr Cécile Münch-Alligné, Dr Christophe Nicolet et Dr Philippe Kobel pour leurs conseils avisés lors de leurs visites régulières au LMH. Je remercie également le Dr Loïc Andolfatto pour le modèle paramétrique de l'auget et j'espère qu'il se convertira à l'utilisation de *SPHEROS* tout prochainement.

Je remercie les étudiants des 20 projets de semestre ou de master pour lesquels j'ai participé à l'encadrement. En particulier, je remercie Felipe Mendes, Guilherme Barros, Julien Bonfils, Filippo Contestabile, Thomas Crevoisier et Amin Niayifar d'avoir contribué au développement de *SPHEROS*.

Un grand merci à tous les collègues doctorants qui m'ont côtoyé durant ma thèse. Je les remercie pour leur soutien que ce soit professionnel, culinaire, sportif ou récréatif. Les parties endiablées de beach-volley ou l'élaboration des, désormais célèbres, pâtes à la saucisse resteront des moments inoubliables de ma thèse au LMH. Merci pour tous les excellents moments de convivialité passés ensemble lors des différentes défenses publiques ou de mes sorties à Lausanne. Un merci tout particulier à ceux que j'ai réussi à faire sortir de la capitale pour diverses activités dans la campagne vaudoise.

En particulier, je remercie Sébastien Alligné et Vlad Hasmatuchi pour les bons moments de détente passés lors de la conférence *SimHydro* à Nice. Merci à Steven Roth pour les discussions sur la passerelle. Un grand merci à Marc Tinguely pour les parties de tennis et à Martino Reclari et Olivier Pacot pour leur coaching au volley et leurs fréquentes envies de pizza à la Puccia. Je remercie également Andres Müller pour ses conseils avisés pour l'organisation de mes voyages et sa tolérance envers mon caractère suisse-allemand. Merci à Ebrahim Jahanbakhsh, créateur de *SPHEROS*, de m'avoir initié aux méthodes particulaires; ses conseils ont été précieux pour la réussite de ma thèse. Je le remercie également pour les bons moments passés ensemble lors des quatre conférences *SPHERIC* auxquelles nous avons participé.

Je remercie Christian Landry et Matthieu Dreyer qui me supportent depuis le début de mes études à l'EPFL. Merci à Christian pour son expertise valaisanne lors des soirées raclette et sa disponibilité pour nos nombreuses discussions que ce soit pour nos thèses ou l'assistanat. Merci à Matthieu pour sa créativité dans l'achat de matériel pour la guerre de bureau. Merci à Arthur Favrel pour ses performances de chanteur en fin de soirée. Bon courage à ces trois derniers pour la rédaction de leurs thèses. Je souhaite également bonne chance à Keita Yamamoto, Simon Pasche, Outi Supponen, Quentin Fabien, Elena Vagnoni, Joao Delgado et Emmanuel Dorel pour la réussite de leur doctorat respectif.

Merci à mes collègues Christophe, Olivier, Andres, Ebrahim, Christian, Matthieu, Arthur et Keita qui m'ont accompagné à la conférence de l'*IAHR* à Montréal. Malgré mon orteil cassé, c'était une conférence grandement réussie et une excellente découverte des bons restaurants et bars à Montréal. Un grand merci à Andres, Matthieu, Arthur et Keita pour le roadtrip à travers le Québec après la conférence.

Je remercie aussi l’équipe du char des Bonzes des Brandons de Payerne, la société de jeunesse de Corcelles-près-Payerne, ainsi que les sociétés de tir de Vuarens et Corcelles-près-Payerne qui m’ont permis de me changer les idées de temps à autre en participant avec grand plaisir à leurs diverses activités.

Enfin, je remercie également mes parents, Michèle et Gérald, qui m’ont toujours soutenu dans le choix de mes études. Merci à mon frère, François, de m’avoir motivé à effectuer un doctorat. Et pour terminer, je tiens à remercier tout particulièrement mon épouse, Mélanie, pour son soutien tout au long de ces quatre années de thèse. Merci de m’avoir motivé à accomplir ce travail.

Villars-le-Terroir, le 14 décembre 2014 Christian Vessaz

Abstract

The prediction of hydraulic machines performances is of high interest to manufacturers in today's highly competitive market for new development or refurbishment of hydraulic power plant. An accurate prediction of the machines performances by numerical simulation allows to reduce the time required for the design phase. In order to predict the resulting torque of a Pelton turbine, the physics of the free jet has to be modeled accurately. Indeed, the deviation of the high-speed water jet is the key phenomenon, which produces the wall pressure field on the buckets and defines the trajectories of the water sheets in the casing. The purpose of this Doctoral Thesis is to develop and define the methodology for new numerical simulations, which capture accurately the deviation of high-speed jet flows.

The use of particle-based methods is investigated instead of using conventional grid-based methods. The advantage of particle-based methods is their Lagrangian formulation, which avoids the well-known difficulties of the mesh generation for complex geometries with moving interfaces. Finite Particle Method (FPM) and Finite Volume Particle Method (FVPM) are used to improve the overall accuracy of the simulations compared to standard Smoothed Particle Hydrodynamics (SPH). The drawback of particle-based methods is their significant increase of computational costs compared to conventional grid-based methods. To mitigate this drawback, the simulations are performed with the FPM/FVPM solver *SPHEROS* developed at EPFL for massively parallel simulations on the *Lemanicus* BG/Q supercomputer. A new adaptive domain decomposition strategy is proposed to perform efficient highly parallelized simulations.

The development of the FPM/FVPM solver is validated by comparing its results with experimental data and conventional grid-based simulations for different test cases. First, the impinging jet on a flat plate validates that FPM and FVPM are able to capture accurately the free surface location as well as the pressure profile on the flat plate at different impinging angles. Second, the steady bucket analysis highlights the convergence of the FVPM results according to the spatial discretization. Finally, the rotating buckets analysis shows that the pressure field in the buckets inner wall is in good agreement with the experimental and numerical data and the evolution of the relative flow pattern matches the flow high-speed visualization. Moreover, the FVPM simulation is able to capture the pressure peak during the impingement first stage, which is also highlighted in the measurements.

Keywords: numerical simulation, particle-based method, free surface flow, free jet, Pelton turbine, bucket

Résumé

La prédiction de la performance des machines hydrauliques présente un grand intérêt pour les constructeurs afin qu'ils puissent s'adapter à la demande dans un marché hautement concurrentiel que ce soit pour un nouveau projet ou pour la rénovation de centrale hydroélectrique. Une prédiction précise des performances par simulation numérique permet de réduire le temps requis pour la phase de design. Afin de prédire le couple résultant d'une turbine Pelton, la physique du jet libre doit être modélisée avec précision. En effet, la déviation du jet d'eau à grande vitesse est le phénomène qui génère le champ de pression sur les augets et définit la trajectoire des nappes d'eau. Le but de cette thèse de doctorat est de développer de nouvelles simulations numériques qui capturent précisément la déviation d'un jet libre.

L'utilisation de méthodes particulaires est privilégiée par rapport aux méthodes conventionnelles à base de maillage. La formulation lagrangienne des méthodes particulaires évite les difficultés de la génération de maillage pour des géométries complexes avec interfaces mobiles. Afin d'améliorer la précision des simulations, par rapport à la méthode standard *Smoothed Particle Hydrodynamics* (SPH), les méthodes des particules finies (FPM) et des particules à volume fini (FVPM) sont utilisées malgré leur coût de calcul élevé. Pour remédier à cet inconvénient, les simulations sont effectuées avec le solveur FPM/FVPM *SPHEROS* développé à l'EPFL pour les simulations massivement parallèles sur le supercalculateur *Lemanicus* BG/Q. Une nouvelle stratégie de décomposition automatique du domaine de calcul est proposée afin d'effectuer efficacement ces simulations massivement parallèles.

Le développement du solveur FPM/FVPM est validé par différents cas d'étude, en comparant les résultats obtenus avec des données expérimentales et des simulations numériques à base de maillage. Premièrement, le jet impactant sur une plaque plane valide que les méthodes FPM et FVPM sont capables de capturer avec précision l'emplacement de la surface libre, ainsi que le profil de pression sur la plaque plane pour différents angles d'impact. Deuxièmement, l'analyse de l'auget fixe met en évidence la convergence des résultats FVPM selon la discrétisation spatiale. Finalement, l'analyse des augets tournant montre que le champ de pression sur la paroi intérieure de l'auget concorde avec les données expérimentales et numériques et que l'évolution de la surface libre correspond à la visualisation à grande vitesse de l'écoulement. De plus, la simulation FVPM est capable de capturer le pic de pression au début de l'impact entre le jet et l'auget, ce qui est également mis en évidence par les mesures.

Mots-clés : simulation numérique, méthode particulaire, écoulement à surface libre, jet libre, turbine Pelton, auget

Contents

Acknowledgements **v**

Abstract (English/Français) **ix**

List of figures **xvii**

List of tables **xxiii**

Nomenclature **xxv**

1 Introduction **1**
- 1.1 Pelton turbine 1
- 1.2 State of the art 4
  - 1.2.1 Numerical simulations of free jet flows and Pelton buckets 4
  - 1.2.2 Particle-based methods 5
- 1.3 Thesis objective and methodology 8
- 1.4 Document structure 9

2 Computational Model **11**
- 2.1 Governing equations 11
  - 2.1.1 Pelton turbines 11
  - 2.1.2 Inviscid weakly compressible flow 13
- 2.2 Finite Particle Method (FPM) 14
  - 2.2.1 Formulation 14
  - 2.2.2 Discretization 15
  - 2.2.3 Introduction of diffusive terms 16
  - 2.2.4 Time integration 17
  - 2.2.5 Shifting strategy 17
  - 2.2.6 Boundary conditions 19
- 2.3 Finite Volume Particle Method (FVPM) 21
  - 2.3.1 Formulation 21
  - 2.3.2 Discretization 22
  - 2.3.3 Inviscid flux 24
  - 2.3.4 Time integration 24
  - 2.3.5 Particle velocity correction 25

2.3.6 Boundary conditions 26
2.4 SPHEROS 27
2.4.1 Main features and applications 27
2.4.2 Octree neighbor search 28
2.4.3 Parallelization 30
2.4.4 Solver implementation 32

3 Adaptive Domain Decomposition 35
3.1 Recursive multi-section algorithm 35
3.2 Adaptive process 36
3.3 Validation 38
3.3.1 Orbital shaking 38
3.3.2 Pelton buckets 44

4 Impinging Jet on a Flat Plate 49
4.1 Case study 49
4.2 FPM validation 51
4.2.1 Shifting strategy 51
4.2.2 Inlet conditions 52
4.2.3 Artificial viscosity 54
4.3 Results 54
4.3.1 Pressure profile 54
4.3.2 Free surface location 55
4.3.3 FPM vs FVPM simulations 55

5 Impinging Jet on Pelton Buckets 61
5.1 Steady bucket analysis 61
5.1.1 Case study 61
5.1.2 Convergence study 62
5.1.3 Results 64
5.2 Rotating buckets analysis 70
5.2.1 Case study 70
5.2.2 Convergence study 71
5.2.3 Results 75
5.3 Free surface evolution analysis 88
5.3.1 Free surface reconstruction 88
5.3.2 Convergence study 88
5.3.3 Results 93

6 Conclusions and Perspectives 99
6.1 Conclusions 99
6.2 Perspectives 100

A Parametric Model of a Pelton Bucket 103

Bibliography **119**

Curriculum Vitae **121**

List of Figures

1.1 Appropriate operating range for different runner types: available water head in function of the specific speed [8]. 2
1.2 Outline of an injector and a Pelton runner. 2
1.3 External flow visualization (left) and test rig casing with plexiglas window (right). Figures obtained from [65]. 3

2.1 Outline of a Pelton bucket. 12
2.2 The shifting strategy restores a more uniform particles distribution. At the end of each time step, a small displacement is applied to all the particles. These displacements depend on the particles location to ensure that the particles do not spread artificially through the free surface. 18
2.3 Outline of the solid boundary, represented by three layers of dummy particles under the solid limit. The boundary velocity of the dummy particle $\boldsymbol{C}_B$ is computed from the ghost velocity $\boldsymbol{C}_G$ in the case of a no-slip wall with a given boundary velocity $\boldsymbol{U}_B$. The distance d_1 is between the dummy particle and the boundary and d_2 between the latter and the ghost sampling location. The normal vector to the boundary is $\boldsymbol{n}_B$ and h is the smoothing length. 20
2.4 Rectangular support kernels and overlapping regions (a) plotted with the contour of Sheppard shape function for the particle i and top-hat kernels (b) [30]. . . . 22
2.5 Outline of the intersection volume between particles i and j [30]. 23
2.6 Weak scaling performed on the *Lemanicus* BG/Q at EPFL. 28
2.7 Schematic subdivision of a computational domain: (a) the Verlet list, (b) the 3-D octree, (c) the complete quadtree and (d) the incomplete quadtree. In (a) and (d), h represents the smoothing length [76]. 29
2.8 Search time measurements for the three search methods [76]. 30
2.9 Domain decomposition and overlapping subdomains [76]. 31

3.1 Uniform domain decomposition (a) and adapted domain decomposition using the recursive multi-section algorithm (b). 36
3.2 Example of a subdomains group for the third main direction. The number of particles in the subdomains are N_A, N_B, N_C and N_D. The interfaces s_1, s_2 and s_3 are moved by the adaptive process. 37

3.3 Example of the interfaces displacements in a subdomains group for the third main direction. The displacements of the interfaces are given by m and $m_{\max}$ denotes the maximum displacement allowed. 38

3.4 Description of the orbital shaking test case. The diameter of the cylinder is D, the initial water height is Z_0 and d_s is the shaking diameter. 39

3.5 Visualization of the *SPHEROS* result using the *POV-Ray* rendering software. . . 39

3.6 Evolution of the free surface on the cylinder's boundary according to the revolutions. Comparison between the measurements from Reclari [71] and the FPM result for a spatial discretization of $Z_0/X_{\text{ref}} = 40$. 40

3.7 Size of the subdomains according to the cylinder's rotations. $L_{\max}$ denotes the maximum load in the subdomains and L_{lim} is the imposed load limit for the ADD process. 41

3.8 Evolution of the maximum load in function of the cylinder's rotations. $L_{\max}$ denotes the maximum load in the subdomains and L_{lim} is the imposed load limit for the ADD process. 42

3.9 Strong scaling analysis of the orbital shaking test case for two different spatial discretizations. The simulations are performed with a fixed uniform domain decomposition (UDD) or with the adaptive domain decomposition process (ADD). 43

3.10 Strong scaling analysis of the orbital shaking test case on the *Lemanicus* BG/Q. The simulations are performed with a fixed uniform domain decomposition (UDD) or with the adaptive domain decomposition process (ADD). 43

3.11 Influence of the domain decomposition on the subdomains sizes. The subdomains sizes are obtained from the converged ADD process on the initial particles position. 45

3.12 Evolution of the maximum load in function of the buckets position. $L_{\max,i}$ denotes the maximum load in the subdomains according to the main direction i and L_{lim} is the imposed load limit for the ADD process. 46

3.13 Computing time for a solver iteration in function of the buckets position. Comparison between simulations with ADD and UDD. 47

4.1 Outline of the impinging jet test case. The Cartesian coordinate system (X, Y and Z) is located on the center of the flat plate. This flat plate can tilt of an angle θ around the Y axis. The inlet of the jet is located at L, D is the jet diameter and $C_{\max}$ is the discharge velocity at the inlet. 50

4.2 Measured velocity profile on the inlet section [41]. The absolute velocity is C, D is the jet diameter and $C_{\max}$ is the discharge velocity at the inlet. 50

4.3 Water jet impinging on the plate at $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$ [38]. 51

4.4 Influence of the particle shifting method on the averaged C_p along the X axis of the flat plate. The pressure peak experiences instabilities when no shifting is applied. The artificial spread of particles through the free surface is highlighted with the shifting vector from Xu et al. [83]. The proposed shifting methods improves the accuracy of the simulation to capture the pressure peak compared to the simulation with the shifting vector from Jahanbakhsh et al. [31]. 52
4.5 Influence of the inlet velocity profile on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$. The uniform velocity profile generates a bell-shaped pressure profile because it does not take into account the disturbance of the flow by the injector. The imposed profile from Kvicinsky et al. [41] fits properly the CFD results and measurements. 53
4.6 Influence of the damping on the time history of C_p at the center of the flat plate. The pressure waves, which are reflected from the plate back to the inlet, are stabilized quicker with the damping zone. 53
4.7 Influence of the artificial viscosity on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$. The artificial viscosity influence is negligible when $\mu^* \leq 0.001$ Pa·s. 54
4.8 Influence of the spatial discretization on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. 56
4.9 Influence of the spatial discretization on the free surface location in the XZ plane for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. 57
4.10 Rendering of the FPM simulations of a $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$ impinging jet using the software *POV-Ray*. 58
4.11 Comparison between FPM and FVPM on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. The spatial discretization corresponds to $D/X_{\text{ref}} = 40$ for both methods. 59

5.1 Outline of the case study. The bucket can tilt of an angle θ around the Z axis. The inlet of the water jet has a diameter D_2 and its axis is in the $-X$ direction at a distance $Y = -D_1/2$. The 32 pressure samples are located on the bucket inner surface, their location fits the position of lines $X1$ to $X7$, $Z1$ to $Z5$ and $D1$ to $D4$. 62
5.2 Time history of the force in a $\theta = 90°$ stationary bucket: influence of the spatial discretization. 63
5.3 FVPM simulation in a stationary bucket at an impinging angle of $\theta = 90°$: free surface reconstruction using *ParaView*. 63
5.4 Averaged C_p along the lines $X1$ to $X7$ of the bucket at an impinging angle of $\theta = 90°$: influence of the spatial discretization. 65
5.5 Averaged C_p along the lines $Z1$ to $Z5$ of the bucket at an impinging angle of $\theta = 90°$: influence of the spatial discretization. 66
5.6 Averaged C_p along the lines $D1$ to $D4$ of the bucket at an impinging angle of $\theta = 90°$: influence of the spatial discretization. 66

5.7 Comparison of the wall pressure field between experimental (up), VOF (middle) and FVPM (down) for the impinging angles $\theta = 72°$. 67
5.8 Comparison of the wall pressure field between experimental (up), VOF (middle) and FVPM (down) for the impinging angles $\theta = 90°$. 68
5.9 Comparison of the wall pressure field between experimental (up), VOF (middle) and FVPM (down) for the impinging angles $\theta = 108°$. 69
5.10 Outline of the case study. The buckets are defined by a reference diameter D_1 and a width B_2. They tilt around the Z axis and their angular position is given by θ. The inlet of the water jet has a diameter D_2 and its axis is in the $-X$ direction at a distance $Y = -D_1/2$. 70
5.11 Outline of the case study. 30 pressure samples are located on the buckets inner wall and 13 on the outer wall. 71
5.12 Evolution of the torque for each bucket (colors) and total torque (black) in function of the angular position. 73
5.13 Averaged torque for one bucket in function of the angular position: influence of the spatial resolution. 73
5.14 FVPM simulation of five rotating buckets: free surface reconstruction using *ParaView*. 74
5.15 Averaged torque for one bucket in function of the angular position: influence of the sound speed. 74
5.16 Time history of C_p at the pressure sensors 1, 2, 3, 6 and 7 on the buckets inner surface, which corresponds to the splitter side of the frontal region of the inner surface. 77
5.17 Time history of C_p at the pressure sensors 12, 13, 14, 15, 20 and 29 on the buckets inner surface, which corresponds to the central region of the inner surface. . . 78
5.18 Time history of C_p at the pressure sensors 4, 5, 8, 9, 10 and 16 on the buckets inner surface, which corresponds to the splitter side of the aft region of the inner surface. 79
5.19 Time history of C_p at the pressure sensors 21, 22, 23, 27, 28 and 30 on the buckets inner surface, which corresponds to the outer side of the aft region of the inner surface. 80
5.20 Time history of C_p at the pressure sensors 17, 18, 19, 24, 25 and 26 on the buckets inner surface, which corresponds to the outer side of the frontal region of the inner surface. 81
5.21 Time history of C_p at the pressure sensors 31, 32, 33, 34, 35 and 36 on the buckets outer surface, which corresponds to the splitter edge of the outer surface. . . . 82
5.22 Time history of C_p at the pressure sensors 38, 39, 40, 41, 42 and 43 on the buckets outer surface, which corresponds to the outer edge of the outer surface. 83
5.23 Pressure field on the bucket inner wall for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°$ and $\theta = 72°$. 84
5.24 Pressure field on the bucket inner wall for the impinging angles of $\theta = 77°, \theta = 82°, \theta = 87°$ and $\theta = 94°$. 85

5.25 Pressure field on the bucket inner wall for the impinging angles of $\theta = 99°, \theta = 104°, \theta = 109°$ and $\theta = 104°$. 86
5.26 Pressure field on the bucket inner wall for the impinging angles of $\theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$. 87
5.27 Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°$ and $\theta = 72°$. The spatial discretization investigated are $D_2/X_{\mathrm{ref}} = 20$ (left) and $D_2/X_{\mathrm{ref}} = 40$ (right). 89
5.28 Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 77°, \theta = 82°, \theta = 87°$ and $\theta = 94°$. The spatial discretization investigated are $D_2/X_{\mathrm{ref}} = 20$ (left) and $D_2/X_{\mathrm{ref}} = 40$ (right). 90
5.29 Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 99°, \theta = 104°, \theta = 109°$ and $\theta = 114°$. The spatial discretization investigated are $D_2/X_{\mathrm{ref}} = 20$ (left) and $D_2/X_{\mathrm{ref}} = 40$ (right). 91
5.30 Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$. The spatial discretization investigated are $D_2/X_{\mathrm{ref}} = 20$ (left) and $D_2/X_{\mathrm{ref}} = 40$ (right). 92
5.31 Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°$ and $\theta = 72°$. 94
5.32 Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 77°, \theta = 82°, \theta = 87°$ and $\theta = 94°$. 95
5.33 Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 99°, \theta = 104°, \theta = 109°$ and $\theta = 114°$. 96
5.34 Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$. 97

A.1 General dimensions defined in [4]. 104
A.2 Physical points considered to describe the inner surface [4]. 105
A.3 Definition of the inlet orientation angle ι [4]. 105
A.4 Control points defining the first Bézier surface [4]. 107
A.5 Control points defining the second Bézier surface [4]. 107
A.6 Control points defining the third Bézier surface [4]. 108
A.7 Definition of the fourth vertex of the third Bézier surface with respect to physical points [4]. 108
A.8 Control points defining the fourth Bézier surface [4]. 109
A.9 Variable thickness map defining the outer surface as an offset of the inner surface [4]. 110
A.10 Representation of half a bucket [4]. 110
A.11 Illustration of the heeling phenomenon [4]. 112

List of Tables

2.1 Complexity of different search algorithms [76]. 29

3.1 Influence of the domain decomposition on the maximum load in the main directions and mean particles number per core. 44

4.1 Influence of spatial and time discretizations. 55

5.1 Influence of the spatial discretization. 62
5.2 Influence of the spatial discretization. 72

A.1 Inventory of the parameters describing the Pelton bucket geometry. 111

Nomenclature

Acronyms

ADD	Adaptive Domain Decomposition
ALE	Arbitrary Lagrangian-Eulerian
CFL	Courant Friedrichs Lewy
EPFL	École polytechnique fédérale de Lausanne
FDM	Finite Difference Method
FEM	Finite Element Method
FPM	Finite Particle Method
FVM	Finite Volume Method
FVPM	Finite Volume Particle Method
I/O	Input and Output
MPI	Message Passing Interface
SPH	Smoothed Particle Hydrodynamics
VM	Vortex Method
VOF	Volume Of Fluid

Latin letters

a	sound speed	$[\mathrm{m\ s^{-1}}]$
B_2	bucket width	[m]
c_2	overall efficiency of the injector	[-]
C	absolute flow velocity	$[\mathrm{m\ s^{-1}}]$
$\boldsymbol{C}$	absolute flow velocity vector	$[\mathrm{m\ s^{-1}}]$
Cu	peripheral component of the absolute flow velocity	$[\mathrm{m\ s^{-1}}]$
D	diameter	[m]
$\boldsymbol{D}$	deformation rate tensor	$[\mathrm{s^{-1}}]$
E	available specific energy	$[\mathrm{J\ kg^{-1}}]$
g	gravity	$[\mathrm{m\ s^{-2}}]$
$\boldsymbol{g}$	gravity vector	$[\mathrm{m\ s^{-2}}]$
h	smoothing length	[m]
H	head	[m]

$\boldsymbol{I}$	idendity matrix	[-]
m	mass	[kg]
$\boldsymbol{n}$	normal vector	[-]
N	number of particles	[-]
p	static pressure	[N m^{-2}]
P	power	[W]
Q	discharge	[m^{3} s^{-1}]
$\boldsymbol{s}$	deviatoric stress tensor	[N m^{-2}]
t	time	[s]
Δt	time step	[s]
T	torque	[N m]
U	peripheral flow velocity	[m s^{-1}]
V	volume	[m^{3}]
W	relative flow velocity	[m s^{-1}]
X	Cartesian coordinate	[m]
$\boldsymbol{X}$	position vector	[m]
X_{ref}	reference particle spacing	[m]
Y	Cartesian coordinate	[m]
z_0	number of injectors	[-]
z_b	number of bucket	[-]
Z	Cartesian coordinate	[m]

Finite Particle Method (FPM) Latin letters

A	damping term for the momentum equation	[m s^{-2}]
d_1	distance between dummy particles and boundary interface	[m]
d_2	distance ghost sampling and boundary interface	[m]
E	right hand side of the mass equation	[kg m^{-3} s^{-1}]
f	test function	[*]
F	right hand side of the momentum equation	[m s^{-2}]
L	blending function	[-]
M_1	first damping term for the mass equation	[kg m^{-3} s^{-1}]
M_2	second damping term for the mass equation	[kg m^{-3} s^{-1}]
$\boldsymbol{R}$	shifting vector	[-]
$\tilde{\boldsymbol{R}}$	intermediary shifting vector	[-]
S	Sheppard coefficient	[-]
W	kernel function	[m^{-3}]
$\tilde{W}$	renormalized kernel function	[m^{-3}]

Finite Volume Particle Method (FVPM) Latin letters

$\boldsymbol{B}$	boundary interaction vector	[m^{-2}]
$\boldsymbol{F}$	flux function vector	[*]

$\boldsymbol{G}$	inviscid contribution of the flux function vector	[*]
$\boldsymbol{P}$	pressure contribution of the flux function vector	[*]
$\boldsymbol{Q}$	viscous contribution of the flux function vector	[*]
$\boldsymbol{R}$	correction coefficient for mass conservation	[N s m^{-3}]
$\boldsymbol{S}$	Free surface area vector	[m^{2}]
$\Delta\boldsymbol{S}$	surface vector	[m^{2}]
$\boldsymbol{U}$	conserved variables vector	[*]
W	kernel function	[m^{-3}]
$\dot{X}$	particle velocity vector	[m s^{-1}]
$\dot{X}'$	velocity correction vector	[m s^{-1}]

Greek letters

β	relative flow angle	[°]
$\boldsymbol{\Gamma}$	interaction vector	[m^{-2}]
Δ	loss factor in the bucket	[-]
$\boldsymbol{\Delta}$	weight vector	[m^{-2}]
η	efficiency	[-]
θ	rotation angle	[°]
λ	blending function	[-]
μ^{*}	artificial dynamic viscosity	[N s m^{-2}]
ξ	FPM Laplacian coefficient	[m^{-5}]
ρ	density	[kg m^{-3}]
σ	kernel summation function	[m^{-3}]
$\boldsymbol{\sigma}$	stress tensor	[N m^{-2}]
ψ	shape function	[m^{-3}]
ω	angular velocity	[rad s^{-1}]
Ω	computational domain	[m^{3}]

Subscripts

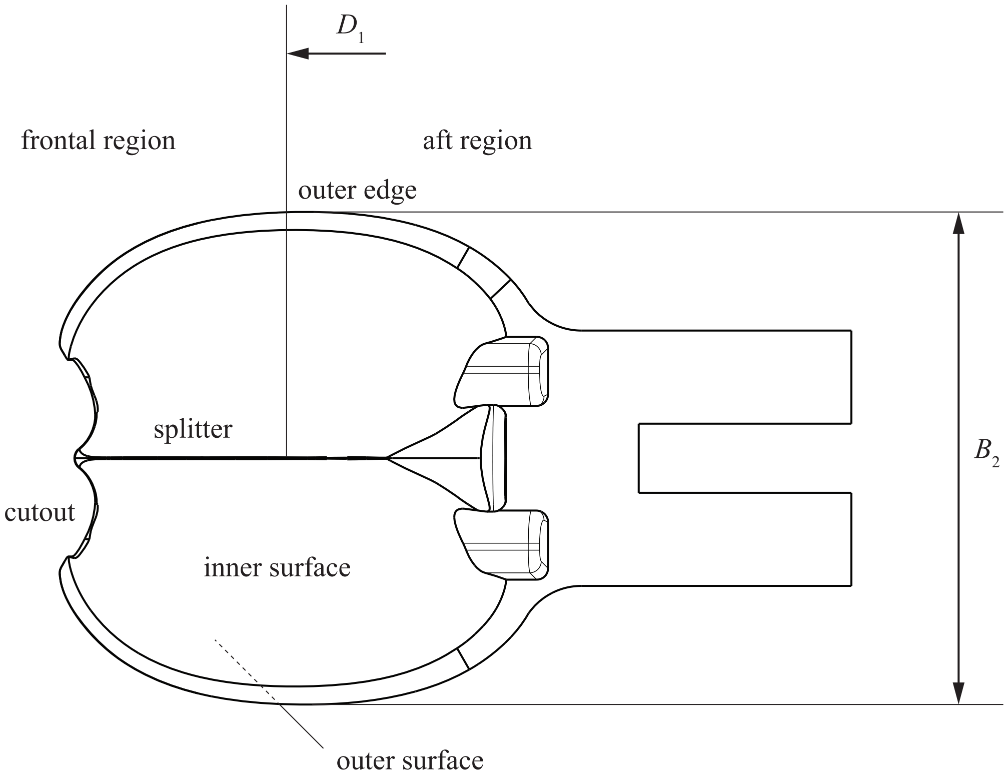
1	runner inlet reference section
$\bar{1}$	runner outlet reference section
2	jet reference section
B	boundary
G	ghost
h	hydraulic
i	particle i
j	particle j
max	maximum
ref	reference
t	transferred

α	Cartesian coordinate α
β	Cartesian coordinate β
$,\alpha$	derivative according to Cartesian coordinate α
$,\beta$	derivative according to Cartesian coordinate β

Non-dimensional variables

φ_{B2}	discharge coefficient; $\varphi_{B2} = 4Q/U_1 z_0 \pi B_2^2$	[-]
ψ_1	energy coefficient; $\psi_1 = 2E/U_1^2$	[-]
C_p	pressure coefficient; $C_p = p - p_{\text{ref}}/0.5\rho_{\text{ref}} C_{\text{max}}^2$	[-]

Bucket geometric definitions



1 Introduction

1.1 Pelton turbine

Over the past few decades, the electricity consumption has significantly increased, which raises many concerns about the productions methods and locations. Furthermore, environmental issues tend to limit or replace the use of non-renewable energy such as fossil or nuclear fuels. In this context, hydropower has become an essential component of the electricity generation and grid stabilization. Hydropower features many different technologies: run-of-the-river, storage, pumped-storage, tidal or wave power plants. According to the available water head and discharge, a different type of turbine is selected. The appropriate operating range of different runners is presented in Figure 1.1 based on statistics for many large hydro power plants. Moreover, each runner geometry is adapted according to the operating conditions in order to maximize the efficiency of the power generation. The present study is focussed on the Pelton turbine, which is mainly used for high head and low discharge power plant.

Lester Pelton developed in 1879 a water wheel featuring several double half-cylindrical buckets and patented it in 1889. These buckets split and reverse a high-speed water jet, which generates a mechanical torque applied on the water wheel. Around 1895, William Doble improves the shape of the Pelton buckets introducing an elliptical shape and a cutout. The cutout provides to the water jet a cleaner bucket entry and gives to the Pelton runner the appearance it still has today. An outline of the Pelton turbine is given in Figure 1.2. The Pelton turbine is an impulse rotating machine where one or several injectors convert the available water head into high-speed water jets. The deviation of the high-speed jets by the buckets converts the kinetic energy of the flow into tangential forces. These forces applied on the buckets produces a torque on the shaft. To produce the electrical power, an electrical generator is linked to the Pelton runner shaft. The intensity of the torque is highly dependent on the shape of the bucket, and especially, on its outlet angle and cutout. Indeed, the outlet angle of the flow has to be as close as possible to 180 degrees to maximize the jets deviation but the tail water should also not hit the backside of the next bucket. Moreover, the location of the cutout, which defines the leading edge of a bucket, should maximize the averaged force on the rotating bucket.

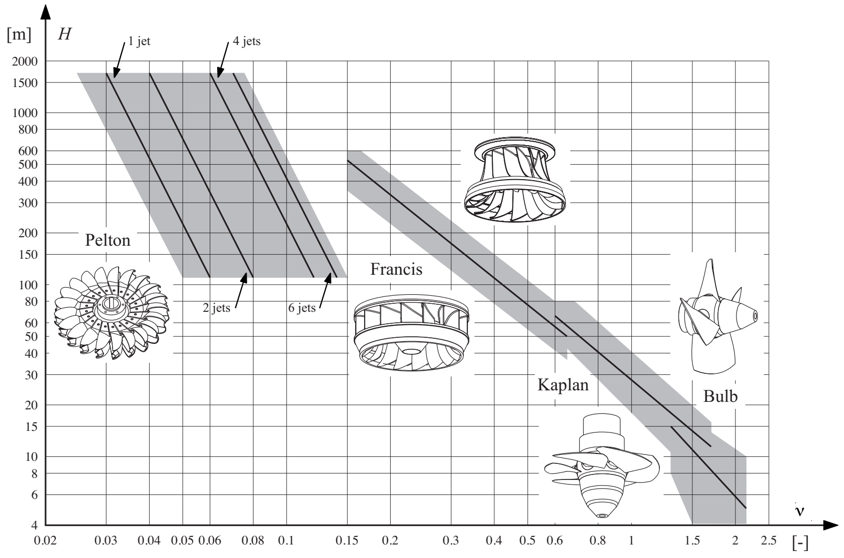


Figure 1.1 – Appropriate operating range for different runner types: available water head in function of the specific speed [8].

Consequently, the design of the Pelton buckets has to be adapted to the available water head and discharge to improve the overall efficiency of the power generation. The thickness of the buckets root is designed according to the material fatigue caused by the frequent passage of the buckets through the high-speed water jets. The design of the buckets geometry also includes further considerations to increase the operating hours of the runner e.g. avoiding the appearance of cavitation and diminishing the effect of the erosion due to silt particles loaded in the flow.

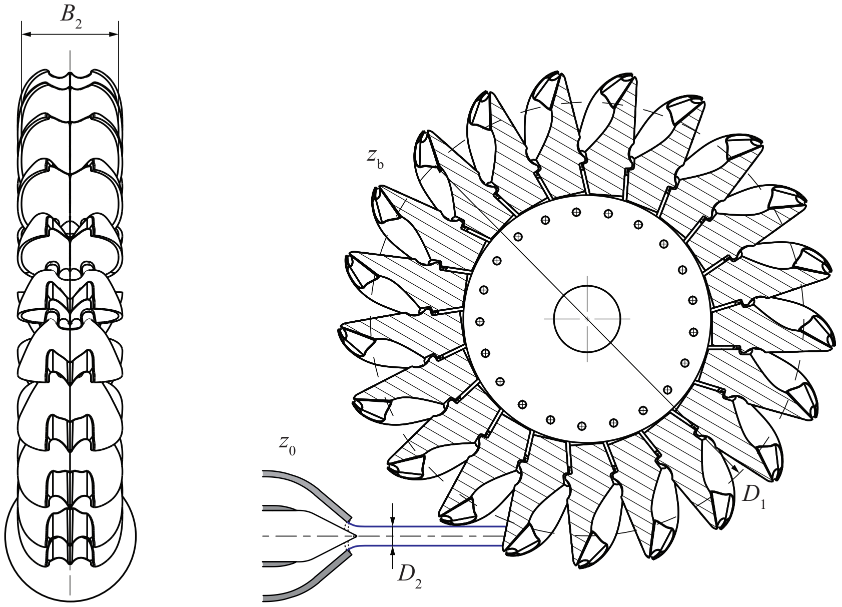


Figure 1.2 – Outline of an injector and a Pelton runner.

The design process of Pelton turbines unit, is difficult compared to reaction turbines. It is usually conducted from know-how and extensive experimental testing on reduced-scale physical models. The transposition from scaled-down models to prototype has to be fine tuned because the flow in a Pelton unit is composed of four different regimes: confined steady-state flows in the upstream pipes and distributor, free-jet flows past the injectors, transient free-surface flows in the buckets, and two-phase dispersed flows in the casing. Each regime features a different characteristic length, a different velocity scale, and is therefore dominated by different forces [65]. These different flows configurations are highlighted in Figure 1.3. An external flow visualization presents the impingement of the high-speed water jet on the buckets cutout and splitter. The water is deviated by the buckets inner surface and escape the bucket forming several water sheets. The test rig casing with plexiglas window shows that these water sheets saturate the air in the casing. Besides this harsh environment, the lack of space between the buckets makes flow visualization difficult to perform [69, 67].

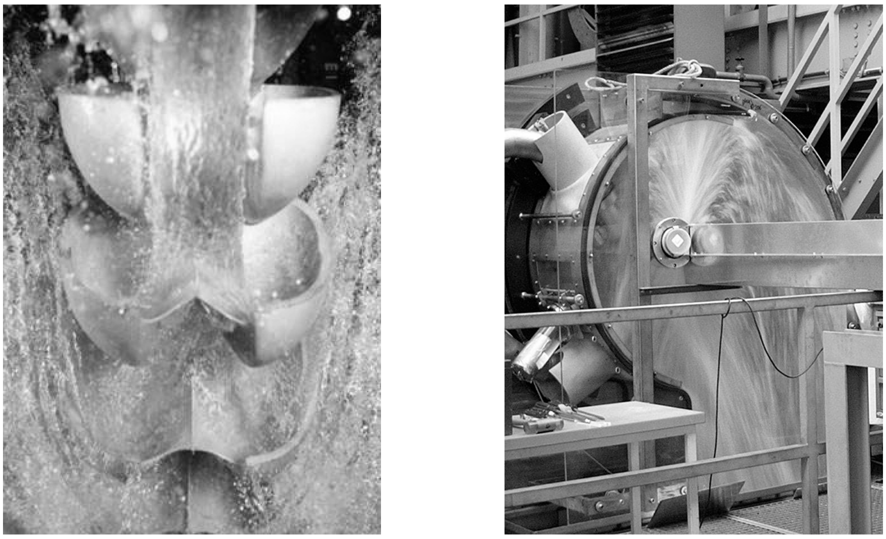


Figure 1.3 – External flow visualization (left) and test rig casing with plexiglas window (right). Figures obtained from [65].

The prediction of the hydraulic machine performance is of high interest to manufacturers in today highly competitive market for new development or refurbishment. An accurate prediction of the machines performance by numerical simulation allows to reduce the time required for the design phase. Therefore, the physics of the free jet has to be modeled accurately. Indeed, the deviation of the high-speed water jet is the key phenomenon, which produces the wall pressure field on the buckets inner surfaces and defines the trajectories of the water sheets.

1.2 State of the art

1.2.1 Numerical simulations of free jet flows and Pelton buckets

The physics of free jet at high Reynolds number is important for many industrial applications and, in particular, for hydropower. The first numerical simulations of this type of flow appeared in the 1990s thanks to the improvement of the computational resources and to the development of the Volume Of Fluid (VOF) method by Hirt and Nichols [27] in 1981. VOF is a grid-based Eulerian method, which tracks the location of the interface between two fluids. A volume fraction function is used to determine if the grid cells are empty, partially-field or full of the traced fluid. The physical properties in the cells are computed separately using the Navier-Stokes equations. The formulation of the VOF method is mass conservative. Consequently, VOF can handle simulations of break apart or joined interfaces.

Numerical simulations of the flow in the Pelton distributor and injectors were performed using VOF by Parkinson et al. [64] to analyzed the effect of the distributor on the velocity profile inside the water jets. Mack and Moser [49] showed the importance of grid refinement to capture accurately the flow at the outlet of a Pelton injector nozzle, and so, to obtain the correct water jet diameter. This effect was also highlighted by Jost et al. [36] in the simulation of a Pelton distributor and two injectors using the two-phase homogeneous model. This model is a simplification of VOF where the mixture is considered as a whole and is resolved in a macroscopic point of view. This assumption is applied because the system does not include phase changes, i.e. negligible drift or diffusion of mass [65]. The deviation of a high-speed water jet by a flat plate at different impinging angles was investigated by Kvicinsky et al. [41, 39] to validate that VOF is able to capture the deviation of the jet. In this study the numerical simulations are validated with pressure measurements on the plate and a water layer thickness comparison. The importance to impose the proper velocity profile at the inlet is highlighted. Kvicinsky [38] measured the velocity profile in the water jet using a Pitot tube. The impinging jet test case was also used for the validation of the particle-based numerical methods from Marongiu et al. [52, 50], Antuono et al. [5] and Molteni et al. [56].

The simulations of steady Pelton buckets at different impinging angles were first computed using VOF [9, 25]. Janetzky et al. [35] and Chaudhari et al. [13] discretized the simulation of a rotating Pelton runner into several steady simulations. These studies have highlighted the importance to simulate accurately the wall pressure field in the buckets inner surfaces. Moreover, Guilbaud et al. [24] showed that a 2-D steady bucket simulation cannot be used to simplify the flow simulation in the Pelton buckets. Kvicinsky et al. [39] and Zoppe et al. [86] compared VOF results to pressure sensors located on the bucket inner surface. These comparisons show that the force on the steady bucket is well captured. However, the experimental flow visualizations of Zoppe et al. [86] highlight that VOF underestimates the leakage through the cutout. Marongiu et al. [53] used the same case study as Kvicinsky et al. [39] to validate their particle-based solver.

The deviation of a high-speed water jet by the rotating Pelton buckets is a challenging fluid mechanics problem, which involves complex geometries, moving boundaries, free surface flows and high-pressure variations. The ability to simulate accurately the instantaneous wall pressure field in the Pelton buckets is a key issue for the design of Pelton runners [49]. Kvicinsky et al. [40] and Perrig et al. [68] simulated the flow in the rotating buckets using the two-phase homogenous model. Perrig et al. [66] compared the two-phase homogenous model to measurements and flow visualization. This study highlights different zones in the Pelton bucket, which contribute differently to the torque generation. Santolin et al. [72] used the two-phase homogenous model to compute the time history of the torque and analyzed the influence of the water jet profile. Jost et al. [36] highlighted the grid influence on the efficiency prediction as well as the needs of significant computing resources to compute accurately a Pelton runner using the two-phase homogenous model. Xiao et al.[81] performed a VOF simulation of a rotating Pelton runner. The computed efficiency is a bit lower than the experiments and the dependance on the grid resolution is also highlighted. Marongiu et al. [51, 54] demonstrated that particle-based methods are well suited to compute the flow in a Pelton runner. However, particle-based methods require significant computational ressources. Anagnostopoulos and Papantonis [3] and Xiao et al. [82] proposed a fast Lagrangian computation to design Pelton buckets. However, this method is only based on the inlet and outlet velocity vectors of the particles, which provides an estimation of the integrated pressure. Neither the whole pressure field nor the exact water sheet location can be accurately computed.

1.2.2 Particle-based methods

Nowadays, the significant increase of the computational power allows the development of new numerical methods to perform more complex and realistic simulations. Particles-based methods, or meshless methods, have been investigated following the needs of various fields of research e.g. astrophysics, engineering or film industry. In opposition to grid-based methods i.e. Finite Difference Method (FDM), Finite Volume Method (FVM) or Finite Element Method (FEM), particle-based methods do not require to know the connectivity between the computing nodes, which is provided by the mesh. Particle-based methods use a set of arbitrarily distributed nodes, so called particles, to solve the motion equations. The material velocity is usually applied to the particles, which allows the use of a Lagrangian formulation. The motion of nodes enables these methods to simulate moving interfaces problems without issues due to mesh deformation or tangling e.g. free-surface, deformable boundary, moving interface or extremely large deformation [44].

Particle-based methods may be categorized in two types of discretization: strong and weak forms. In the strong form methods, the partial differential equations are discretized at the particles center using shape functions to obtain a set of discretized equations. Example of these methods include: the general finite difference method [15], the hp-meshless cloud method [43], the meshless collocation method [84], the Vortex Method (VM) [37] and the Smoothed Particle Hydrodynamics (SPH) [58]. In VM and SPH methods, the approximation is performed using a weak form but the equations are discretized at the particles positions. The common property of the weak form methods is that the shape functions are constructed entirely using a set of nodes in a locally supported domain, and so, no element is needed for the shape functions [30]. Example of these methods include: the diffuse element method [62], the element free Galerkin [11], the radial point interpolation method [80], the reproducing kernel particle method [47], the meshless local Petrov-Galerkin [7], the moving least squares particle hydrodynamics [17] and the Finite Volume Particle Method (FVPM) [26]. The advantage of strong form methods like SPH is their ease of implementation. However, SPH suffers from a lack of consistency and the boundary conditions enforcement or multi-scale resolution are still open challenges. The advantages of the weak form methods are that they are consistent and conservative regardless of any variation in particles resolution and they discretize the boundary conditions precisely. However, these methods require a significant computational cost to integrate the shape functions.

The standard SPH method was developed in 1977 by Lucy [48], and Gingold and Monaghan [21] for astrophysical simulations. The advantage of this method is its Lagrangian formulation, which avoids the difficulties of the mesh generation for complex geometries with moving interfaces. However, this method suffers from a lack of accuracy compared to the grid-based numerical simulations when dealing with non-uniform particles distribution. Therefore, many different renormalized SPH methods were developed to restore the consistency e.g. Rieman SPH [79, 53] or Incompressible SPH (ISPH) [83]. However, these renormalized methods present an increase of computational time due to the resolutions of the Poisson equation or Riemann system. Chen and Beraun [14] and Liu et al. [46], proposed a set of correction formulas for the SPH kernel function, which they called the Finite Particle Method (FPM). Comparing to standard SPH, the FPM requires more computational costs to compute the renormalized kernel function and derivatives but it ensures the consistency of the simulation even for a non-uniform particles distribution.

During the SPH simulations, the contraction of the streamlines as well as the tensile instability described by Monaghan [59] result in the clustering of particles. This particles clustering increases the spatial discretization error, which decreases the overall accuracy of the simulation. To restore a better particles distribution, Xu et al. [83] apply a small shift to each particle at the end of each time step. This particle shifting strategy has demonstrated its efficiency for confined flows. Recently, Jahanbakhsh et al. [31] extend this method for free surface flows.

The solid wall boundary treatment in SPH is handled by replacing the wall boundary surface with so called "boundary particles". Three main approaches are found in the literature. First, the repulsive force approach described by Monaghan and Kajtar [57] or Marongiu et al. [52] consists in placing boundary particles on the wall surface. These boundary particles exert a repulsive force, which is proportional to the distance between the boundary and the fluid. However, the particle spacing of the boundary has to be fine-tuned to a smaller value compared to the particle spacing in the fluid domain, the tuning depending on the flow conditions. The second approach used by Bierbrauer et al. [12] places "mirroring ghost particles" behind the wall boundary surface. The drawback of this second approach is that a particular attention to the generation of the mirroring particles has to be given close to the sharp edges to ensure a proper boundary treatment. The third boundary treatment is based on fixed ghost particles as described by Adami et al. [1] and Marrone et al. [55] where "dummy particles" are placed under the wall boundary with a fixed location. The physical properties of theses dummy particles are then updated from a ghost sampling in the fluid region at each time step.

FVPM is a particle-based solver introduced by Hietel [26] in 2000. In 2009, Nestor et al. [63] extended the method to incompressible flows. This method features an Arbitrary Lagrangian-Eulerian (ALE) formulation, which means that the computing nodes can either moves with the material velocity or a user-prescribed velocity. This method is able to satisfy free surface and no-slip wall boundary conditions precisely. FVPM combines attractive features of SPH and conventional grid-based FVM. Like SPH, FVPM is based on a kernel and a smoothing length to compute the interactions between the particles. Like FVM, this method is consistent because it uses the interaction vectors to weight the fluxes exchanged between the particles, which overcomes the main drawback of standard SPH. Moreover, FVPM is locally conservative regardless of any variation in particles smoothing length [70]. This enables users to refine the solution by splitting the particles in the region of interest and perform accurate simulation more efficiently [34]. In FVPM, the control volumes are replaced by particles and the exchange occurs through the interfaces defined by overlapping regions. For each pair of overlapping particles, two interaction vectors are defined. Their difference is analogous to the area vector in FVM. Due to the complexity of shape functions, their integrations are usually approximated using Quadrature rules over a large number of integration points. In 2011, Quinlan and Nestor [70] developed a new FVPM in which the integrals are computed quickly and exactly for 2-D simulations. They simplified the shape functions to circular top-hat kernels and achieved a reasonable compromise between computational cost and accuracy. Recently, Jahanbakhsh et al. [33] introduced rectangular top-hat kernels to compute exactly the integrals in 3-D.

The main drawback of all the particle-based methods is the significant increase of computational cost compared to grid-based methods. Therefore, the development of massively parallel codes on multi-CPU or multi-GPU clusters is required [74] to mitigate this drawback. According to Holmes et al. [28] and Basa et al. [10], the use of multi-CPU clusters with an efficient communication allows to reach a near linear speed up. However, a particular attention has to be paid to obtain a uniform load on each CPU. As the computational domain is split in multiple subdomains, and each subdomain is assigned to a single core, the computational time is linked to the most loaded subdomain. Therefore, the size of the subdomains have to be adapted during the simulation [16]. In the literature, two main strategies have been used for the Adaptive Domain Decomposition (ADD) of particle-based methods. The first one is based on the space filling curves, which map the 3-D space into a 1-D curve. Springel [73] used this strategy for cosmological simulations. The second strategy is based on the recursive multi-section algorithm, recently improved by Ishiyama et al. [29].

1.3 Thesis objective and methodology

Efficient Pelton turbine engineering requires accurate flow numerical simulation to compute the net torque acting on the Pelton runner. This resulting torque is produced by the deviation of the high-speed water jets by the buckets. Therefore, the unsteady wall pressure field on the buckets needs to be accurately simulated to yield, after integration, the time history of the runner torque. The purpose of this Doctoral Thesis is to develop and define the methodology for new numerical simulations, which capture accurately the deviation of high-speed jet flows. The development of consistent numerical methods are investigated to simulate the flow in rotating Pelton buckets. Highly-parallelized simulations using dynamic load balancing are implemented to reduce the computing time and improve the efficiency of the computations. The assessment of the simulations is performed by comparing the physics of the flow to available numerical and experimental data.

The use of particle-based methods is investigated to perform the numerical simulations. The advantage of these methods is their Lagrangian formulation, which avoids the well-known difficulties of the mesh generation for complex geometries with moving interfaces. Moreover, particle-based methods are well suited to simulate free-surface flows subject to extremely large deformation [44]. For the numerical simulations, the water flow is assumed to be single phase, without surface tension and weakly compressible. The state equation of the water flow is used to derive the pressure from the actual density, by passing the need of solving a Poisson equation if a strictly incompressible flow were assumed [2]. The weakly compressible flow hypothesis imposes that the Mach number remains always below 0.1 all along the numerical simulation, which allows to increase the time step value compared to a pure compressible simulation.

The use of FPM is investigated to improve the overall accuracy of the simulations compared to standard SPH thanks to its renormalized kernel. A new particle shifting method is proposed to stabilize the simulation and avoid the artificial spread of particles through the free surface. The solid boundary is modeled by three layers of dummy particles and an improved boundary treatment is applied to ensure the proper location of the interface. The impingement of a high-speed water jet on a flat plate is simulated using FPM to validate that the jet deviation is captured accurately. The FPM results are compared with experimental data and VOF numerical simulations from Kvicinsky et al. [41].

The use of FVPM is investigated to simulate high-speed jet flow impinging on Pelton buckets. In FVPM, the wall boundary is represented by one layer of particles located on the bucket surface, which overcome the difficulty of FPM to model accurately the geometry of the splitter. Another advantage of FVPM is that its formulation is consistent and conservative. First, the flow in a steady bucket is simulated using FVPM to validate that the wall pressure field on the bucket inner surfaces matches the experimental and VOF results from Kvicinsky et al. [39]. Second, the simulation of rotating Pelton buckets is simulated using FVPM and the results are compared to pressure measurements and flow visualization from Perrig [65].

The numerical simulations are performed with the FPM/FVPM solver *SPHEROS*, developed at EPFL since 2010 [32]. This software is designed for massively parallel computing using MPI library and tree-based neighbor searching. To reduce the computing time, the computational domain is split into several subdomains. An adaptive domain decomposition strategy based on the recursive multi-section algorithm of Ishiyama et al. [29] is proposed to make the load in the subdomains uniform. Consequently, efficient highly-parallelized simulations can be run on the *Lemanicus* BG/Q at EPFL.

1.4 Document structure

This Doctoral Thesis is organized in five chapters. Chapter 1 presents the introduction including an overview of Pelton turbines and high-speed jet flows, a literature review of grid-based and particle-based numerical simulations of high-speed jet flows as well as the Thesis objective and methodology.

Chapter 2 includes the fundamental theory of Pelton turbines and the governing equation of the fluid flow. Then, this chapter describes the computational models used for the FPM and FVPM solvers, including the discretization, particle shifting strategy and solid boundary condition. Finally, the main characteristics of the *SPHEROS* software are introduced.

Chapter 3 presents the proposed adaptive domain decomposition strategy. First, the orbital shaking test case is used to validate the efficiency of the ADD strategy. A strong scaling analysis on the *Lemanicus* BG/Q at EPFL shows the performance of the ADD. Second, the simulation of rotating Pelton buckets highlights the importance of load balancing for simulations featuring highly scattered particles.

Chapter 4 includes the validation of the FPM to simulate the deviation of high-speed water jet on a flat plate. First, the effects of the particle shifting strategy and inlet treatment are discussed. Then, a convergence study of the FPM results is presented for three different impinging angles. The pressure profile on the flat plate as well as the free surface location are compared to VOF and measurement from Kvicinsky et al. [41]. Finally, a comparison of FPM and FVPM results is given for the three different impinging angles.

Chapter 5 is devided in three sections. The first section presents the FVPM simulation of a steady bucket at three different impinging angles. A convergence study analyses the effect of spatial discretization on the wall pressure field inside the bucket. Next, the wall pressure field is compared to the VOF and measurements of Kvicinsky et al. [39]. The second section, contains the rotating Pelton buckets analysis. The spatial discretization is validated by analyzing the time history of the torque applied on the buckets for different particles resolutions. The wall pressure field in the buckets inner and outer surfaces are compared to the VOF results and experimental data of Perrig [65]. Finally, the evolution of the relative flow pattern inside the buckets are compared to the VOF results and experimental data in the third section.

2 Computational Model

2.1 Governing equations

2.1.1 Pelton turbines

The Pelton turbine is an impulse hydraulic machine as described in Figure 1.2. The available specific energy E is converted by the z_0 injectors into high-speed water jets with a velocity

$$C_2 = c_2\sqrt{2E} = c_2\sqrt{2gH} \tag{2.1}$$

where H is the head, g the gravity and c_2 the overall efficiency of the injector. The latter is assumed constant and usually included in $0.98 < c_2 < 0.99$. The jet diameter D_2 is imposed by the needle stroke, which controls the discharge Q through the Pelton unit.

The z_b buckets deviate the high-speed water jets, which generates a mechanical torque. The velocity triangles at the inlet (subscript 1) and outlet (subscript $\bar{1}$) of a Pelton bucket are sketched in Figure 2.1. In this figure, C represents the absolute flow velocity, U the peripheral flow velocity and W the relative flow velocity. The transferred specific energy is obtained by the Euler's equation for hydraulic turbomachine

$$E_t = U_1 \cdot Cu_1 - U_{\bar{1}} \cdot Cu_{\bar{1}}. \tag{2.2}$$

By assuming that the mean flow enters and leaves the buckets at the reference diameter D_1 of the Pelton runner, the peripheral flow velocity is equal to

$$U_1 = U_{\bar{1}} = \omega D_1/2 \tag{2.3}$$

where ω is the angular speed. Using the relation $W_1 = C_1 - U_1$ from the inlet velocity triangle and introducing the loss factor in the bucket $\Delta = 1 - W_{\bar{1}}/W_1$, the Euler's equation becomes

$$E_t = U_1 W_1 \left[1 + (1-\Delta)\cos(\beta_{\bar{1}})\right]. \tag{2.4}$$

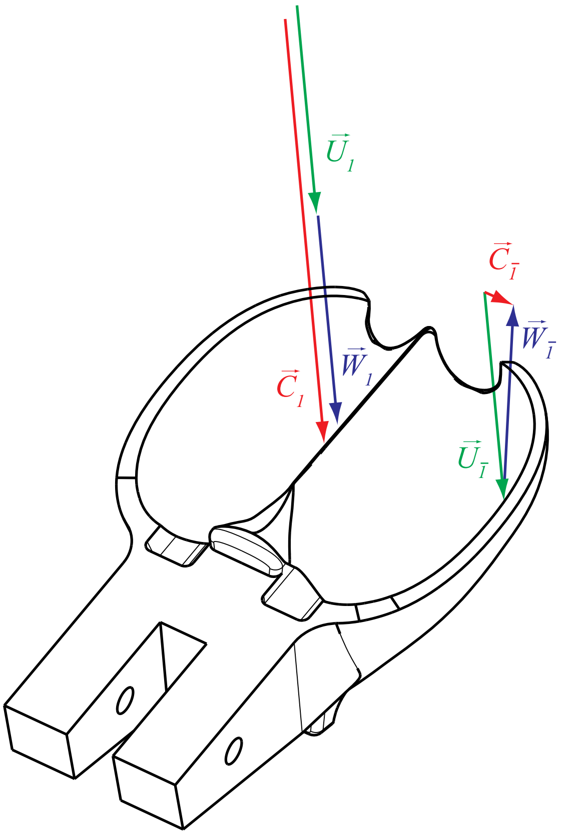


Figure 2.1 – Outline of a Pelton bucket.

The expression of the transferred power is given by

$$P_t = \rho Q E_t = \rho Q U_1 (C_1 - U_1) \left[1 + (1 - \Delta)\cos(\beta_{\bar{1}})\right] \tag{2.5}$$

and the torque is expressed by

$$T_t = \frac{P_t}{\omega} = \frac{\rho Q D_1}{2} (C_1 - U_1) \left[1 + (1 - \Delta)\cos(\beta_{\bar{1}})\right]. \tag{2.6}$$

The theoretical maximum transferred power is obtained for $U_1 = C_1/2$. The runaway condition is fulfilled when $U_1 = C_1$, i.e. the torque and transferred power become zero.

The characteristic of a Pelton runner is described by an hillchart in function of the discharge and energy coefficients. These coefficients are respectively

$$\varphi_{B2} = \frac{Q}{Q_{\text{ref}}} = \frac{4Q}{U_1 z_0 \pi B_2^2} \tag{2.7}$$

and

$$\psi_1 = \frac{2E}{U_1^2}. \tag{2.8}$$

The hillchart represents the hydraulic efficiency of the Pelton unit in function of the operating condition. The hydraulic efficiency is expressed by

$$\eta = \frac{P_t}{P_h} = \frac{P_t}{\rho Q E}. \tag{2.9}$$

2.1.2 Inviscid weakly compressible flow

The water flow is assumed inviscid and weakly compressible. The flow motion is governed by the mass and linear momentum conservation equations

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{C} \tag{2.10}$$

and

$$\frac{d\left(\rho \boldsymbol{C}\right)}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{2.11}$$

where ρ is the density, $\boldsymbol{C}$ the velocity vector, $\boldsymbol{g}$ the gravity vector and $\boldsymbol{\sigma} = \boldsymbol{s} - p\boldsymbol{I}$ the stress tensor, which includes p the static pressure and $\boldsymbol{s}$ the deviatoric stress contribution defined as

$$\boldsymbol{s} = 2\mu^* \left(\boldsymbol{D} - \frac{1}{3}\mathrm{tr}\left(\boldsymbol{D}\right)\boldsymbol{I}\right). \tag{2.12}$$

In the case of an inviscid flow, the deviatoric stress contribution is equal to zero. However, in the present study, an artificial viscosity μ^* is introduced to damp the numerical oscillations. In the latter expression, $\boldsymbol{D}$ is the deformation rate tensor

$$\boldsymbol{D} = \frac{\nabla \boldsymbol{C} + (\nabla \boldsymbol{C})^T}{2}. \tag{2.13}$$

In order to close the system of equations (2.10) and (2.11), the pressure is derived from the state equation

$$p = \frac{\rho_{\mathrm{ref}} a^2}{7} \left(\left(\frac{\rho}{\rho_{\mathrm{ref}}} \right)^7 - 1 \right) \tag{2.14}$$

where ρ_{ref} is the reference density and a is the sound speed. According to the weakly compressible assumption, the sound speed is set to $10 \cdot C_{\mathrm{max}}$, C_{max} being the discharge velocity of the water jet. This assumption is based on the weakly compressible approach of Monaghan [60] to ensure that the density variations remain below one percent and the Mach number remains below 0.1 all along the numerical simulation. Therefore, the weakly compressible approach allows to increase the time step value compared to a pure compressible simulation by decreasing the sound speed in the CFL condition. Moreover, the weakly compressible approach does not need to solve a Poisson equation, which is time consuming in the strictly incompressible approach [2].

2.2 Finite Particle Method (FPM)

2.2.1 Formulation

The standard SPH formulation of a function is based on a kernel approximation and a decomposition of the continuous matter into a set of N disordered particles. Therefore, the function

$$f\left(\boldsymbol{X}_i\right)=\sum_j^N f\left(\boldsymbol{X}_j\right) W\left(\boldsymbol{X}_i-\boldsymbol{X}_j, h\right) V_j \tag{2.15}$$

is evaluated for each particle i by its j neighboring particles. The kernel function W weights the interaction between these particles according to a given smoothing length h. The advantage of this formulation is that the derivatives

$$\frac{\partial f\left(\boldsymbol{X}_i\right)}{\partial X_\alpha}=-\sum_j^N f\left(\boldsymbol{X}_j\right) \frac{\partial W\left(\boldsymbol{X}_i-\boldsymbol{X}_j, h\right)}{\partial X_\alpha} V_j \tag{2.16}$$

are directly computed from the derivatives of the kernel [45].

Each particle features a uniform mass m_i and a volume

$$V_i=\frac{m_i}{\rho_i} \tag{2.17}$$

where ρ_i is the density of the particle. The distance between two particles is defined by the vector

$$\boldsymbol{X}_{ij}=\boldsymbol{X}_i-\boldsymbol{X}_j \tag{2.18}$$

whose components in the Cartesian coordinate system are X_α with $\alpha=[X, Y, Z]$. To simplify the notation, the approximated function f and kernel W are expressed as

$$f_i=f\left(\boldsymbol{X}_i\right); \qquad f_{i,\alpha}=\frac{\partial f_i}{\partial X_\alpha} \tag{2.19}$$

and

$$W_{ij}=W\left(\boldsymbol{X}_{ij}, h\right); \qquad W_{ij,\alpha}=\frac{\partial W_{ij}}{\partial X_\alpha}. \tag{2.20}$$

The Laplacian of the function $\boldsymbol{\nabla}^2 f$ is computed according to Morris et al. [61] in order to conserve the linear momentum:

$$\boldsymbol{\nabla}^2 f_i=\sum_j^N \xi_{ij}\left(f_i-f_j\right) V_j \tag{2.21}$$

where

$$\xi_{ij} = \frac{\frac{1}{2}\left(\tilde{W}_{ij,\alpha} - \tilde{W}_{ij,\alpha}\right) X_{ij,\alpha}}{\left\|\boldsymbol{X}_{ij}\right\|^2 + 10^{-4} h^2}. \tag{2.22}$$

In the present work, the quintic Wendland kernel used by Fatehi and Manzari [20] is chosen

$$W_{ij} = \frac{21}{2\pi h^3} L\left(\boldsymbol{X}_{ij}\right) \left(1 - \frac{\|\boldsymbol{X}_{ij}\|}{h}\right)^4 \left(4\frac{\|\boldsymbol{X}_{ij}\|}{h} + 1\right); \tag{2.23}$$

$$L\left(\boldsymbol{X}_{ij}\right) = \begin{cases} 1 & \text{if} \, \|\boldsymbol{X}_{ij}\| < h, \\ 0 & \text{otherwise.} \end{cases} \tag{2.24}$$

This kernel is used with a constant smoothing length $h = 2.6 X_{\text{ref}}$ where X_{ref} is the reference particle spacing.

The derivatives of the standard SPH formulation are not consistent for a non-uniform particle distribution. Therefore, the FPM formulation proposed by Chen and Beraun [14] and Liu et al. [46] is applied. This approach is based on the Taylor series expansions of the function f, which is weighted by the SPH kernel

$$\sum_j^N f_j W_{ij} V_j = f_i \sum_j^N W_{ij} V_j + f_{i,\alpha} \sum_j^N X_{ji,\alpha} W_{ij} V_j \tag{2.25}$$

and its derivatives

$$\sum_j^N f_j W_{ij,\beta} V_j = f_i \sum_j^N W_{ij,\beta} V_j + f_{i,\alpha} \sum_j^N X_{ji,\alpha} W_{ij,\beta} V_j \tag{2.26}$$

to compute the renormalized FPM kernel

$$\left[\begin{array}{c} \tilde{W}_{ij} \\ \tilde{W}_{ij,\beta} \end{array} \right] = \left[\begin{array}{cc} \sum_j^N W_{ij} V_j & \sum_j^N X_{ji,\alpha} W_{ij} V_j \\ \sum_j^N W_{ij,\beta} V_j & \sum_j^N X_{ji,\alpha} W_{ij,\beta} V_j \end{array} \right]^{-1} \left[\begin{array}{c} W_{ij} \\ W_{ij,\beta} \end{array} \right]. \tag{2.27}$$

2.2.2 Discretization

The water flow is assumed weakly compressible and inviscid. The hydrodynamics equations are built according to the energy-based framework of Fang et al. [19]. The flow motion is governed by the governing equations (2.10) and (2.11), which are discretized following the FPM formulation:

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j^N \left(C_{i,\beta} - C_{j,\beta}\right) \tilde{W}_{ij,\beta} V_j \tag{2.28}$$

and

$$\rho_i \frac{DC_{i,\alpha}}{Dt} = \sum_j^N \left[p_i \tilde{W}_{ij,\alpha} + p_j \left(\frac{3}{4} \tilde{W}_{ij,\alpha} - \frac{1}{4} \tilde{W}_{ji,\alpha} \right) \right] V_j + \rho_i g_\alpha. \tag{2.29}$$

The positions of the particles are derived from

$$\frac{DX_{i,\alpha}}{Dt} = C_{i,\alpha} \tag{2.30}$$

while the pressure is derived from the equation of state (2.14).

2.2.3 Introduction of diffusive terms

For reducing the numerical oscillations and stabilizing the simulation, the diffusive terms M_1, M_2 and A are introduced in the mass equation

$$\frac{D\rho_i}{Dt} = E_i \Rightarrow \frac{D\rho_i}{Dt} = E_i + M_{1i} + M_{2i} \tag{2.31}$$

and linear momentum conservation equation

$$\frac{DC_{i,\alpha}}{Dt} = F_{i,\alpha} \Rightarrow \frac{DC_{i,\alpha}}{Dt} = F_{i,\alpha} + A_{i,\alpha}. \tag{2.32}$$

The damping of the spurious oscillations in the pressure field proposed by Fatehi and Manzari [20] is applied to (2.31). The terms

$$M_{1i} = -\rho_i \Delta t \sum_j^N \left(p_{i,\alpha} - p_{j,\alpha} \right) \tilde{W}_{ij,\alpha} V_j; \tag{2.33}$$

$$p_{i,\alpha} = \sum_j^N \left(p_j - p_i \right) \tilde{W}_{ij,\alpha} V_j \tag{2.34}$$

and

$$M_{2i} = 2\rho_i \Delta t \sum_j^N \xi_{ij} \left(p_i - p_j \right) V_j \tag{2.35}$$

represent two different formulations for the approximation of the Laplacian of the pressure. The difference between M_1 and M_2 is used as a numerical filter to damp the pressure oscillations.

The artificial viscosity proposed by Molteni and Colagrossi [56]

$$A_{i,\alpha} = 0.01hC_{\text{ref}}\sum_{j}^{N}\frac{2}{1+\frac{V_j}{V_i}}L\left(\left(C_{i,\beta}-C_{j,\beta}\right)X_{ij,\beta}\right)\xi_{ij}\left(C_{i,\alpha}-C_{j,\alpha}\right)V_j; \tag{2.36}$$

$$L\left(\left(C_{i,\beta}-C_{j,\beta}\right)X_{ij,\beta}\right) = \begin{cases} 1 & \text{if } \left(C_{j,\beta}-C_{j,\beta}\right)X_{ij,\beta} < 0, \\ 0 & \text{otherwise} \end{cases} \tag{2.37}$$

is applied to (2.32) to damp spurious oscillations which appear when SPH is applied to weakly compressible liquid.

2.2.4 Time integration

The time integration of the governing equations is solved with a modified Verlet scheme introduced by Molteni and Colagrossi [56]

$$\left\{\begin{array}{l} C_\alpha^{n+1/2} = C_\alpha^n + \frac{\Delta t}{2}\left(F_\alpha^n + A_\alpha^n\right) \\ X_\alpha^{n+1/2} = X_\alpha^n + \frac{\Delta t}{2}C_\alpha^n \\ \rho^{n+1/2} = \rho^n + \frac{\Delta t}{2}\left(E^n + M_1^n + M_2^n\right) \end{array}\right. \Rightarrow \left\{\begin{array}{l} C_\alpha^{n+1} = C_\alpha^n + \Delta t\left(F_\alpha^{n+1/2} + A_\alpha^{n+1/2}\right) \\ \Downarrow \\ X_\alpha^{n+1} = X_\alpha^{n+1/2} + \frac{\Delta t}{2}C_\alpha^{n+1} \\ \Downarrow \\ \rho^{n+1} = \rho^{n+1/2} + \frac{\Delta t}{2}\left(E^{n+1} + M_1^{n+1} + M_2^{n+1}\right) \end{array}\right. \tag{2.38}$$

This modified Verlet scheme evaluates the variables at the midtime $n+1/2$, which corresponds to the time step $\Delta t/2 = \left(t^{n+1}-t^n\right)/2$. Then, the velocities are computed at the next time step $n+1$ and used to update the positions of the particles. Finally, the density is evaluated according to the new velocities and positions.

During the simulation, the time step Δt is updated to fulfill the Courant Friedrichs Lewy (CFL) criteria defined as

$$(C_{\text{ref}} + C_{\max})\frac{\Delta t}{h} \leq \max(\text{CFL}) \tag{2.39}$$

2.2.5 Shifting strategy

At the end of each time step, the particle shifting applies a small displacement $\delta\boldsymbol{X}$ to all the particles. This displacement is computed by

$$\delta\boldsymbol{X}_i = 0.0125C_{\max}\Delta t\boldsymbol{R}_i \tag{2.40}$$

where $\boldsymbol{R}_i$ is the shifting vector, which depends on the location of the neighboring particles. The updated values of velocity and pressure are extrapolated from the particles displacements knowing the gradients at the former particles positions.

The purpose of the shifting strategy is to restore a more uniform particles distribution. An outline of the shifting strategy is presented in Figure 2.2 for two particles: one being in the bulk flow and the other one being close to the free surface. In the second case a particular attention is paid to avoid an artificial spread of particles through the free surface.

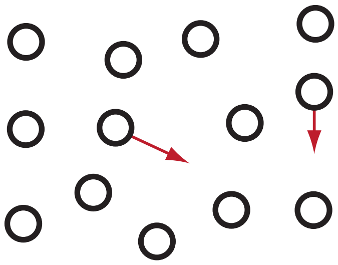


Figure 2.2 – The shifting strategy restores a more uniform particles distribution. At the end of each time step, a small displacement is applied to all the particles. These displacements depend on the particles location to ensure that the particles do not spread artificially through the free surface.

The previous work of Vessaz et al. [78] has shown the importance of using a shifting strategy, which depends on the particle location. For example, the shifting method introduced by Xu et al. [83] is well suited for the bulk flow because it was developed for confined flows and applied to ISPH simulations of a lid-driven cavity. In the Xu et al. [83] method, the shifting vector is given by

$$\boldsymbol{R}_{\mathrm{xu},i} = \sum_{j}^{N} \left(\frac{\bar{r}_i}{r_{ji}} \right)^2 \boldsymbol{n}_{ji}; \tag{2.41}$$

$$\bar{r}_i = \frac{1}{N} \sum_{j}^{N} r_{ji} \tag{2.42}$$

where $r_{ji} = \|\boldsymbol{X}_{ji}\|$ and $\boldsymbol{n}_{ji} = \boldsymbol{X}_{ji} / r_{ji}$ describe respectively the absolute distance and the unit vector between particles i and j. Jahanbakhsh et al. [31] extended the method of Xu et al. [83] for the simulation of a rotating square patch of fluid. To mitigate the artificial spread of particles through the free surface, the neighboring particles used to compute the shifting vector are reduced. In the Jahanbakhsh et al. [31] method, the shifting vector is given by

$$\boldsymbol{R}_{\mathrm{jah},i} = \sum_{j}^{N} \left(\frac{X_{\mathrm{ref}}}{r_{ji}} \right)^2 L\left(X_{\mathrm{ref}} - r_{ji}\right) \boldsymbol{n}_{ji}; \tag{2.43}$$

$$L\left(X_{\mathrm{ref}} - r_{ji}\right) = \begin{cases} 1 & \text{if } X_{\mathrm{ref}} - r_{ji} \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.44}$$

In the present study, a new shifting strategy is followed. The proposed shifting vector is computed in two steps. The first step

$$\tilde{\boldsymbol{R}}_i = \sum_j^N \frac{X_{\text{ref}}^2 + \lambda\left((\bar{r}_i)^2 - X_{\text{ref}}^2\right)}{\left(r_{ji}\right)^2} H\left(X_{\text{ref}} + \lambda\left(h - X_{\text{ref}}\right) - r_{ji}\right) \boldsymbol{n}_{ji} \tag{2.45}$$

corresponds to a linear blending between the former shifting vectors (2.41) and (2.43) according to a blending function λ. The second step

$$\boldsymbol{R}_i = \tilde{\boldsymbol{R}}_i - \left[(1-\lambda)(\tilde{\boldsymbol{R}}_i \cdot \boldsymbol{n}_{FS})\boldsymbol{n}_{FS}\right] \tag{2.46}$$

reduces the shifting according to the normal to the free surface

$$\boldsymbol{n}_{FS} = \frac{\sum_j^N \nabla \tilde{W}_{ij} V_j}{\left\| \sum_j^N \nabla \tilde{W}_{ij} V_j \right\|}. \tag{2.47}$$

The blending function λ provides an estimation of the location of the particles: if $\lambda = 0$, the particle is on the free surface, and if $\lambda = 1$, the particle is in the bulk flow. The blending function is computed by

$$\lambda = L(S_i) \cdot S_{\text{in}}; \tag{2.48}$$

$$L(S_i) = \begin{cases} \frac{\min(S_i,1)-0.8}{0.2} & \text{if}\, S_i > 0.8, \\ 0 & \text{otherwise} \end{cases} \tag{2.49}$$

where S_i is the Shepard coefficient defined as

$$S_i = \sum_j^N W_{ij} V_j. \tag{2.50}$$

The value of S_{in} is an empirical estimation imposed at the particles inlet and kept constant during the simulation. In the present work, the value of S_{in} is linearly interpolated between the jet axis ($S_{\text{in}} = 1$) and the jet surface ($S_{\text{in}} = 0$).

2.2.6 Boundary conditions

Solid boundary

Three layers of dummy particles are located under the solid boundary to ensure that the kernel of the fluid particles is complete. Each dummy particle has a constant mass and reference spacing, which are equal to their corresponding fluid values. The boundary location is defined by the normal vector $\boldsymbol{n}_B$ and the distance d_1. The physical properties of the dummy particles are computed from a ghost sampling in the fluid region, which follows the boundary representation of Adami et al. [1]. An outline of the velocity computation is presented in Figure 2.3 in the case of a no-slip wall.

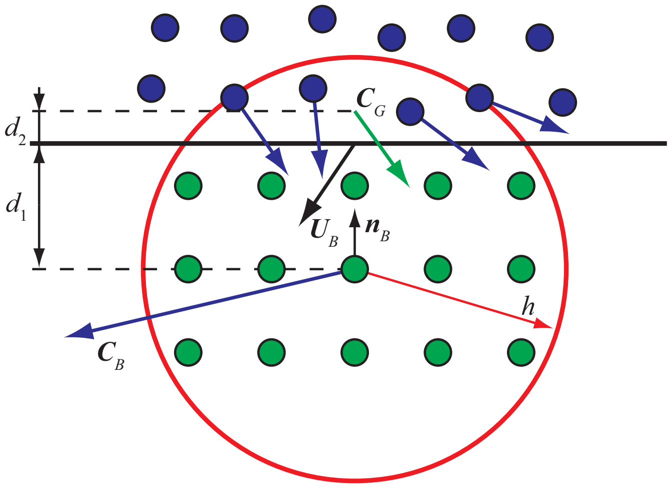


Figure 2.3 – Outline of the solid boundary, represented by three layers of dummy particles under the solid limit. The boundary velocity of the dummy particle $\boldsymbol{C}_B$ is computed from the ghost velocity $\boldsymbol{C}_G$ in the case of a no-slip wall with a given boundary velocity $\boldsymbol{U}_B$. The distance d_1 is between the dummy particle and the boundary and d_2 between the latter and the ghost sampling location. The normal vector to the boundary is $\boldsymbol{n}_B$ and h is the smoothing length.

In this example, the ghost velocity is computed from a Shepard interpolation in the neighboring fluid particles

$$\boldsymbol{C}_{G,i} = \frac{\sum_j^N \boldsymbol{C}_j W_{ij} V_j}{\sum_j^N W_{ij} V_j} \tag{2.51}$$

In order to ensure the proper location of the interface, this velocity is assumed to be imposed at d_2, the distance between the solid wall boundary and the center of the kernel volume in the fluid region

$$d_{2,i} = \frac{3(h - d_{1,i})}{8} \qquad \forall d_{1,i} \in]0, h[\tag{2.52}$$

Consequently, the boundary velocity of the dummy particle is computed by

$$\boldsymbol{C}_{B,i} = \boldsymbol{U}_{B,i} + \frac{d_{1,i}}{d_{2,i}} (\boldsymbol{U}_{B,i} - \boldsymbol{C}_{G,i}) \tag{2.53}$$

which improves the boundary treatment in the case of a no-slip wall with a given boundary velocity $\boldsymbol{U}_{B,i}$.

Inlet boundary with damping zone

The inlet boundary condition is represented by three layers of dummy particles located above the inlet surface to ensure that the kernel of the fluid particles is complete. These inlet particles have a prescribed velocity and move according to the time step value. When an inlet particle crosses the inlet surface, a fluid particle is injected at its location and the inlet particle is moved back of a $3X_{\mathrm{ref}}$ distance. In order to prevent the pressure waves reflection on the inlet surface, a damping zone is applied on the first 20% of the jet length. This damping zone is inspired from the perfectly matched layer method described by Atkins [6]. Inside the damping zone the mass and momentum equations are modified as follow

$$\frac{D\rho_i}{Dt} = \frac{D\rho_i}{Dt} - \left(\rho_i - \rho_{\mathrm{ref}}\right)\frac{\gamma}{\Delta t} \tag{2.54}$$

and

$$\frac{DC_{i,\alpha}}{Dt} = \frac{DC_{i,\alpha}}{Dt} - \left(C_{i,\alpha} - C_{\mathrm{max},\alpha}\right)\frac{\gamma}{\Delta t} \tag{2.55}$$

where γ is the damping coefficient. This coefficient is equal to one at the top and decreases quadratically to zero at the bottom of the damping zone.

2.3 Finite Volume Particle Method (FVPM)

2.3.1 Formulation

The FVPM formulation is based on the Sheppard interpolating or shape function. This function is zero-order consistent and is defined as

$$\psi_i(\boldsymbol{X}) = \frac{W_i(\boldsymbol{X})}{\sigma(\boldsymbol{X})} \tag{2.56}$$

where $W_i(\boldsymbol{X}) = W_i(\boldsymbol{X} - \boldsymbol{X}_i, h)$ is the kernel function and $\sigma(\boldsymbol{X}) = \sum_j W_j(\boldsymbol{X})$ is the kernel summation. The spatial resolution of the interpolation is given by the smoothing length h, which is imposed by the initial spatial resolution $h = 0.75X_{\mathrm{ref}}$. The spatial and temporal derivatives of the Sheppard shape function are expressed by

$$\nabla\psi_i = \frac{\nabla W_i \sigma - \nabla\sigma W_i}{\sigma^2} \tag{2.57}$$

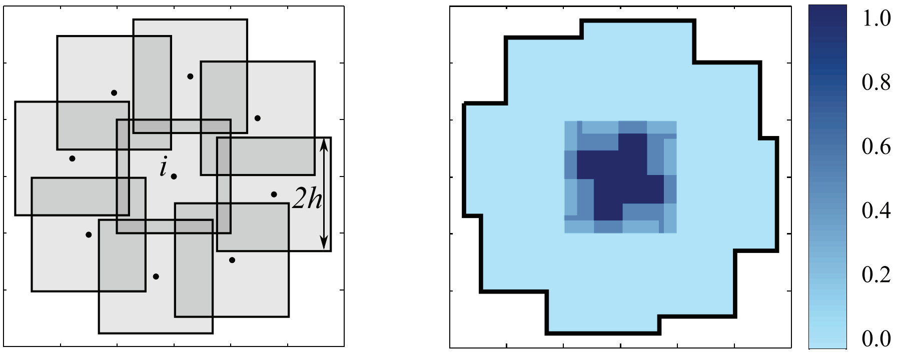
and

$$\frac{\partial W_i}{\partial t} = \frac{\frac{\partial W_i}{\partial t}\sigma - \frac{\partial\sigma}{\partial t}W_i}{\sigma^2} = -\dot{\boldsymbol{X}}_i \cdot \nabla W_i. \tag{2.58}$$

In the present study, a rectangular top-hat kernel is used to compute the interaction vectors, which reads

$$W_i(\boldsymbol{x}) = \begin{cases} 1 & \|\boldsymbol{X}-\boldsymbol{X}_i\|_\infty \le h \\ 0 & \|\boldsymbol{X}-\boldsymbol{X}_i\|_\infty > h \end{cases} \tag{2.59}$$

A 2-D example of particles interactions with rectangular support is given in Figure 2.4(a). The top-hat kernel is less smooth than a bell-shaped kernel as shown by the contours of the Sheppard shape function given in Figure 2.4(b). However, Quinlan and Nestor [70] demonstrated that a top-hat kernel allows a fast and exact computation of the interaction vector in 2-D with a circular support. The rectangular support is used to simplify the geometrical computation in 3-D [33].



(a) (b)

Figure 2.4 – Rectangular support kernels and overlapping regions (a) plotted with the contour of Sheppard shape function for the particle i and top-hat kernels (b) [30].

2.3.2 Discretization

The governing equations (2.10) and (2.11) can be written as the Partial Differential Equation (PDE)

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) = 0 \tag{2.60}$$

where $\boldsymbol{U} = \{\rho, \rho\boldsymbol{C}\}$ represents the conserved variables and $\boldsymbol{F} = \{\rho\boldsymbol{C}, \rho\boldsymbol{C}\boldsymbol{C}-\boldsymbol{\sigma}\}$ is the flux function.

In FVPM, the Sheppard interpolating or shape function ψ is used to discretized the conservative form of the PDE

$$\int_\Omega \frac{\partial \boldsymbol{U}}{\partial t}\psi_i dV + \int_\Omega \nabla \cdot \boldsymbol{F}\psi_i dV = 0 \tag{2.61}$$

where Ω represents the whole computational domain and dV an element of volume. After some mathematical operations [30], the PDE is simplified as

$$\frac{d}{dt}(\boldsymbol{U}_i V_i) + \sum_j \frac{1}{2}\left(\boldsymbol{U}_i + \boldsymbol{U}_j\right)\cdot\left(\dot{\boldsymbol{X}}_i \cdot \boldsymbol{\Gamma}_{ji} - \dot{\boldsymbol{X}}_j \cdot \boldsymbol{\Gamma}_{ij}\right) + \sum_j \frac{1}{2}\left(\boldsymbol{F}_i + \boldsymbol{F}_j\right)\cdot\left(\boldsymbol{\Delta}_{ij} + \boldsymbol{B}_{ij}\right) = 0 \tag{2.62}$$

where $\dot{\boldsymbol{X}}$ is the particle velocity and $\boldsymbol{B}_{ij}$ is the boundary interaction defined by Dilts [18]

$$\boldsymbol{B}_{ij} = \boldsymbol{A}_{ij} + \boldsymbol{A}_{ji} = \int_{\partial\Omega} \psi_j \psi_i \boldsymbol{n} dS. \tag{2.63}$$

In equation (2.62), $\boldsymbol{\Delta}_{ij}$ represents a weight vector

$$\boldsymbol{\Delta}_{ij} = \boldsymbol{\Gamma}_{ij} - \boldsymbol{\Gamma}_{ji} \tag{2.64}$$

which depends on the interaction vector between particles i and j

$$\boldsymbol{\Gamma}_{ij} = \int_{\Omega} \frac{\psi_i \nabla W_j}{\sigma} dV = \int_{\Omega} \frac{W_i \nabla W_j}{\sigma^2} dV. \tag{2.65}$$

Quinlan and Nestor [70] demonstrated that a top-hat kernel allows a fast and exact computation of the interaction vector in 2-D with a circular support. In 3-D, Jahanbakhsh et al. [33] showed that the use of top-hat kernel with a rectangular support reduces significantly the geometrical computations, to compute the integral of (2.65). The latter is simplified as

$$\boldsymbol{\Gamma}_{ij} = -\sum_{l}^{m} \left(\frac{\Delta \boldsymbol{S}_l}{\sigma_l^+ \sigma_l^-} \right) \tag{2.66}$$

where m is the number of partitioned rectangles, $\Delta\boldsymbol{S}$ represents the surface vector of the partitions, σ^- and σ^+ are the summation kernel inside and outside the surfaces respectively. An outline of the 2-D computation of (2.66) is given in Figure 2.5, where the rectangular partitions are reduced to lines segments.

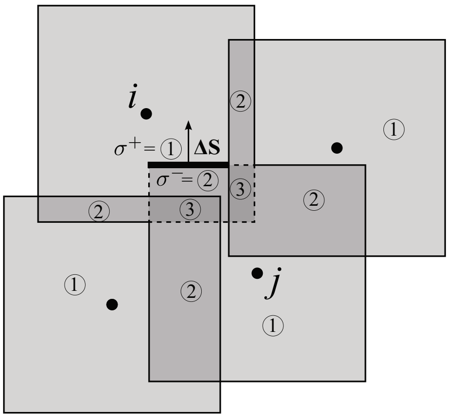


Figure 2.5 – Outline of the intersection volume between particles i and j [30].

2.3.3 Inviscid flux

For reducing the numerical oscillations and stabilizing the simulation, artificial viscosity has to be introduced [58]. In FVPM, the AUSM$^+$ approach followed by Nestor et al. [63] is applied. Therefore, equation (2.62) is rearranged following Jahanbakhsh [30] to exhibit the inviscid terms by decomposing the flux function as

$$\boldsymbol{F} = \boldsymbol{Q} + \boldsymbol{P} - \boldsymbol{G} \tag{2.67}$$

where $\boldsymbol{Q} = \left\{\rho \boldsymbol{C}, \rho \boldsymbol{C}\boldsymbol{C}\right\}$, $\boldsymbol{P} = \left\{0, p\boldsymbol{I}\right\}$ and $\boldsymbol{G} = \{0, \boldsymbol{s}\}$. Equation (2.62) becomes

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + \sum_j \left(\left(\boldsymbol{Q} - \boldsymbol{U}\dot{\boldsymbol{X}}\right)_{ij} + \boldsymbol{P}_{ij}\right) \cdot \boldsymbol{\Delta}_{ij} - \sum_j \boldsymbol{G}_{ij} \cdot \boldsymbol{\Delta}_{ij} = 0. \tag{2.68}$$

In order to damp the spurious numerical oscillations, $\left(\boldsymbol{Q} - \boldsymbol{U}\dot{\boldsymbol{X}}\right)_{ij}$ and $\boldsymbol{P}_{ij}$ are computed using the AUSM$^+$ scheme of Liou [42] and a correction term is applied to the mass flux as described in [30]. The expression of the deviatoric stress $\boldsymbol{G}_{ij}$ is given by

$$\boldsymbol{G}_{ij} = \mu^* \left(\tilde{\nabla} \boldsymbol{C}_{ij} + \left(\tilde{\nabla} \boldsymbol{C}_{ij}\right)^{\mathrm{T}}\right) \tag{2.69}$$

where μ^* is the artificial dynamic viscosity, $\boldsymbol{C}_{ij}$ the averaged velocity between particles i and j and $\tilde{\nabla}$ the gradient operator obtained from weighted least square to avoid double summation of gradient operator [30].

In order to avoid the checker-board oscillations due to pressure and velocities computed at the same computational node, a correction term $\boldsymbol{R}_{ij} \cdot \boldsymbol{\Delta}_{ij}$ is added to the mass conservation equation. The correction coefficient $\boldsymbol{R}_{ij}$ is computed according to Jahanbakhsh [30]

$$\boldsymbol{R}_{ij} = \left(\frac{\nabla p_i + \nabla p_j}{2} - \tilde{\nabla} p_{ij}\right) \Delta t \tag{2.70}$$

where

$$\nabla p_i = \frac{1}{V_i} \sum_j \left(\frac{p_i + p_j}{2}\right) \boldsymbol{\Delta}_{ij} \tag{2.71}$$

and $\tilde{\nabla} p$ is computed at the interface of particles i and j using the gradient operator obtained from weighted least square to avoid double summation of gradient operator.

2.3.4 Time integration

The time integration is performed using a second-order explicit Runge-Kuta scheme. First, the field variables are evaluated at the midtime $n + 1/2$

$$\boldsymbol{U}^{n+1/2} = \boldsymbol{U}^n - \nabla \cdot \boldsymbol{F}\left(\boldsymbol{U}^n\right) \frac{\Delta t}{2} \tag{2.72}$$

which corresponds to the time step $\Delta t/2 = \left(t^{n+1} - t^{n}\right)/2$. Then, the fluxes are computed according to the midtime. Finally, the field variables at the next time step $n+1$ are evaluated using the intermediate fluxes

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^{n} - \nabla \cdot \boldsymbol{F}\left(\boldsymbol{U}^{n+1/2}\right)\frac{\Delta t}{2}. \tag{2.73}$$

The time step value Δt is updated to fulfill the CFL criteria defined as

$$\Delta t = \mathrm{CFL} \cdot \min\left(\frac{h}{a + \|\boldsymbol{C}_i\|}\right). \tag{2.74}$$

2.3.5 Particle velocity correction

FVPM is an ALE method, which means that $\dot{\boldsymbol{x}}_i$, the velocity of the particle i, could be prescribed arbitrarily. In the present study, the fluid particles velocity are set equal to the flow velocity plus a correction vector

$$\dot{\boldsymbol{X}}_i = \boldsymbol{C}_i + \dot{\boldsymbol{X}}_i'. \tag{2.75}$$

The correction vector is defined by

$$\dot{\boldsymbol{X}}_i' = 0.4\left(\frac{h}{C_{\max}}\right)\sum_j \Omega^*_{ij}\frac{\boldsymbol{\Gamma}_{ij} - \boldsymbol{\Gamma}_{ji}}{\left|\boldsymbol{\Gamma}_{ij} - \boldsymbol{\Gamma}_{ji}\right|^2} \tag{2.76}$$

where Ω^*_{ij} represents an interaction volume between particle i and j. Like the particle shifting strategy in FPM, the particle velocity correction is applied to ensure a uniform distribution of particles in the flow and avoid particles clustering. To compute an interaction with another fluid particle, Ω^*_{ij} is defined by

$$\Omega^*_{ij} = V_i \cap V_j \tag{2.77}$$

where $V_i \cap V_j$ represents the intersection volume between the volumes of particle i and j. In the case of an interaction with a wall boundary particle, the interaction volume is increased to prevent the penetration of the fluid particles through the solid boundary. The latter is expressed as

$$\Omega^*_{ij} = 50\left(\frac{2V_{h,i} \cap V_{h,j}}{(2h)^3}\right)^2 \left(V_i \cap V_j\right) \tag{2.78}$$

where $V_{h,i} \cap V_{h,j}$ represents the intersection volume between the kernels of particle i and j.

2.3.6 Boundary conditions

Free surface boundary

One advantage of FVPM compared to FPM is that the free surface particles are known precisely. Indeed, the following condition

$$\boldsymbol{S}_i = \int_{\partial\Omega} \psi_i \boldsymbol{n} dS = \int_{\Omega} \nabla\psi_i dV = \sum_j \boldsymbol{\Delta}_{ij} = 0 \tag{2.79}$$

highlights the interior fluid particles. Therefore, the free surface particles are identified when $\boldsymbol{S}_i$ do not vanish [70] and the normal vector to the free surface is given by

$$\boldsymbol{n}_i = \frac{\boldsymbol{S}_i}{\|\boldsymbol{S}_i\|}. \tag{2.80}$$

In the present study, the pressure of the free surface particles is imposed to zero. To take into account the free surface displacement, Jahanbakhsh [30] expressed the change of volume as

$$\frac{dV_i}{dt} = \sum_j \left(\dot{\boldsymbol{X}}_j \cdot \boldsymbol{\Gamma}_{ij} - \dot{\boldsymbol{X}}_i \cdot \boldsymbol{\Gamma}_{ji}\right) + \dot{\boldsymbol{X}}_i \cdot \boldsymbol{S}_i \tag{2.81}$$

and the impermeability condition as

$$\dot{\boldsymbol{X}}_i \cdot \boldsymbol{n}_i = \boldsymbol{C}_i \cdot \boldsymbol{n}_i. \tag{2.82}$$

To avoid the artificial spread of particles through the interface, the particle velocity correction is modified for all the free surface particles. Therefore, the correction vector (2.76) becomes

$$\dot{\boldsymbol{X}}_i' = \dot{\boldsymbol{X}}_i' - \left(\dot{\boldsymbol{X}}_i' \cdot \boldsymbol{n}_i\right) \boldsymbol{n}_i. \tag{2.83}$$

Solid boundary

To impose the solid boundary condition, one layer of wall boundary particles is located on the interface. In comparison with FPM where three layers of boundary particles are located under the solid boundary, FVPM solid boundary representation is able to model complex geometries exactly e.g. the sharp edge of the bucket splitter. The wall boundary particles have the property of the fluid particles, i.e. their pressure and stress are computed from governing equations of the fluid. However, their velocities $\boldsymbol{C}$ and $\dot{\boldsymbol{x}}$ are imposed equal to the wall velocity to ensure that the wall boundary particles remain fixed on the solid interface. Moreover, no special treatment has to be applied to the governing equations of the fluid particles close to the solid boundary. Indeed, they are considered as interior particles since $\boldsymbol{S}_i = 0$. Consequently, the force applied on the boundary is given by

$$f_{B,i} = \sum_{j\in\text{fluid}} \left(-p_{ij}\boldsymbol{I} + \boldsymbol{s}_{ij}\right)\cdot\boldsymbol{\Delta}_{ij}. \tag{2.84}$$

Inlet boundary

The inlet boundary condition is represented by three layers of dummy particles located above the inlet surface like in FPM. These inlet particles have a prescribed velocity and move according to the time step value. When an inlet particle crosses the inlet surface, a fluid particle is injected at its location and the inlet particle is moved back of a $3X_{\text{ref}}$ distance.

2.4 SPHEROS

2.4.1 Main features and applications

SPHEROS is a particle-based software developed at EPFL since 2010 by Jahanbakhsh et al. [32] and Vessaz et al. [76]. *SPHEROS* stands for SPH and erosion. Its aim is to simulate the silt erosion phenomena occurring in hydraulic machines, and especially in Pelton turbines. This software features SPH, FPM and FVPM solvers for the fluid flow. The simulation of the erosion is performed using the FVPM solver and includes the material deformation, mass removal and rigid spherical silt particles motion [30]. In order to validate this software, different applications and test cases have been investigated:

- rotating square patch and lid-driven cavity [31],
- 2-D impinging jet [32],
- 3-D impinging jet [75, 78],
- orbital shaking [76],
- steady bucket flow [77],
- square cylinder, sedimentation, elastic gate, erosion studies [30],
- and many other e.g. fountain, dam break, water wheel and Pelton injector.

To perform simulations of millions of particles in a reasonable computing time, *SPHEROS* is designed for massively parallel computing. It is developed in C++ to take the advantage of object-oriented programming and use the Message Passing Interface (MPI) library from MPICH2 to parallelize the computation. The parallel Input and Output (I/O) is performed by the HDF5 library to efficiently load and store the results. This library is designed to store, access, analyze, share, and preserve diverse and complex data in heterogeneous computing and storage environments. The parallel I/O driver of HDF5 reduces access time on parallel systems by reading/writing multiple data streams. An octree algorithm is applied to identify the neighbors of each particle at the beginning of each time step. The computational domain is decomposed into a fixed number of subdomains using a recursive multi-section algorithm. The domain decomposition strategy presented in Chapter 3 is applied during the simulations to dynamically balance the particles load in the subdomains.

The efficiency of *SPHEROS* is studied by a weak scaling analysis, which reports the computing time with respect to the number of cores keeping the load per core constant. This analysis is based on FVPM simulations of the liddriven cavity and run on the *Lemanicus* BG/Q at EPFL. The BG/Q is an IBM supercomputer, which features one rack of 1'024 sixteen-cores computing nodes PowerA2 at 1.6 GHz, 16 GB of memory per computing node and a 5D torus communication network. The load per subdomain is set to 25^3 particles, which means that the weak scaling simulations feature between 15'625 and 256'000'000 fluid particles. The evolution of the computing efficiency η is presented in Figure 2.6 between 1 core and the full *Lemanicus* BG/Q i.e. 16'384 cores. The efficiency decreases linearly between 1 and 64 cores. Then, an efficiency around 80% is highlighted between 64 and 16'384 cores.

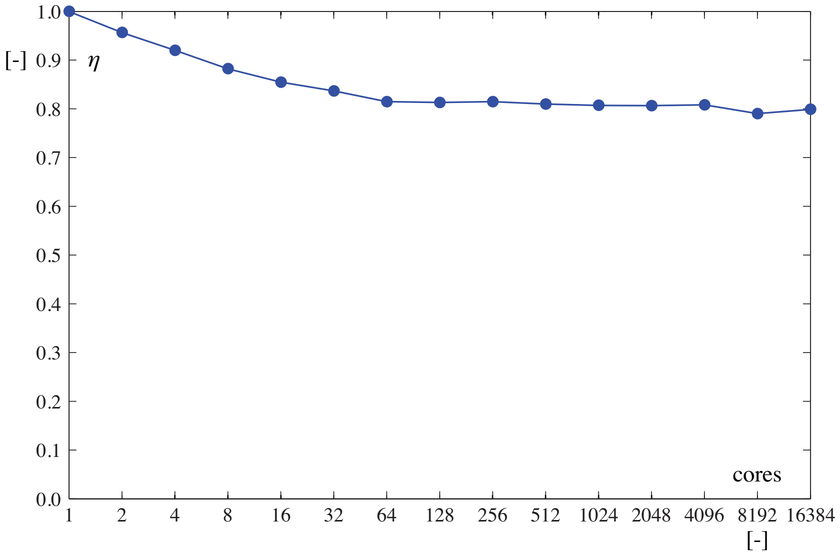
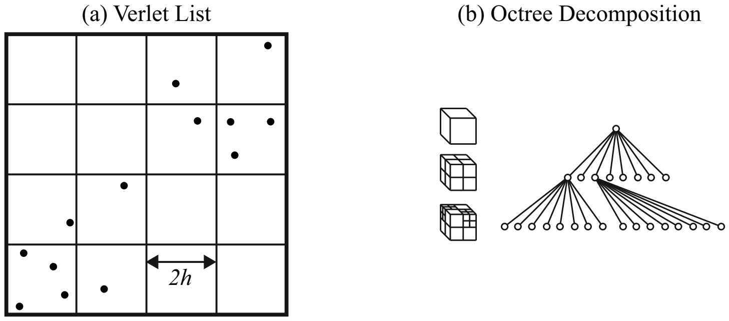


Figure 2.6 – Weak scaling performed on the *Lemanicus* BG/Q at EPFL.

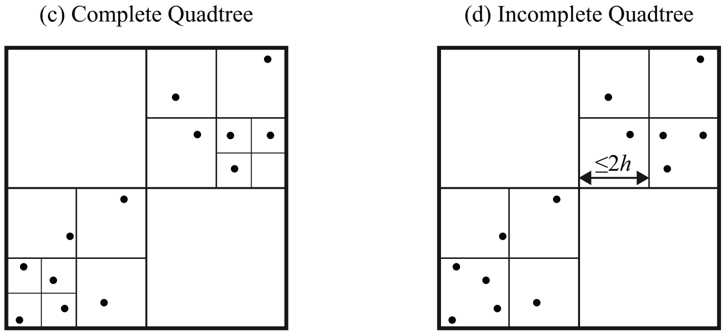
2.4.2 Octree neighbor search

In particle based method, each particle has a close interaction with its neighboring particles, which imposes to identify the neighbors of each particle at the beginning of each time step. However, basic search methods represent a significant computational cost i.e. $O(N^2)$ for N particles. Therefore, more efficient search algorithms such as Verlet list or octree, are investigated.

On the one hand, the Verlet list algorithm is usually chosen in SPH codes [22]. In this method, the computational domain is uniformly divided into numbers of subdivisions, see Figure 2.7(a). In this case, the distance check is performed for a limited number of particles which are placed in the close-enough subdivisions. This method is very efficient for simple particle systems. On the other hand, octree search algorithm [73] is able to adapt subdivisions at the location of the particles. With this method, the root octree recursively splits the maximal problem domain into eight octants, see Figure 2.7(b). The method is called "complete octree" if the branches at the end of the tree contain individual particle [46]. The 2-D complete tree, i.e. quadtree, is shown in Figure 2.7(c). To profit both efficiency and adaptivity, the "incomplete octree" method stops the decomposition when the subdivision size is lower or equal to $2h$, see Figure 2.7(d). Defining M as the average number of particles per subdivision, the construction and access complexities of the Verlet list, complete octree and incomplete octree algorithms are summarized in Table 2.1.


(a) Verlet List

(b) Octree Decomposition


(c) Complete Quadtree

(d) Incomplete Quadtree

Figure 2.7 – Schematic subdivision of a computational domain: (a) the Verlet list, (b) the 3-D octree, (c) the complete quadtree and (d) the incomplete quadtree. In (a) and (d), h represents the smoothing length [76].

Table 2.1 – Complexity of different search algorithms [76].

Complexity	**Verlet list**	**complete octree**	**incomplete octree**
Access	$O(N/M)$	$O(N\log(N))$	$O(N/M\log(N/M))$
Construction	$O(1)$	$O(\log(N))$	$O(\log(N/M))$

To implement the incomplete octree search algorithm in *SPHEROS*, three different methods have been examined. First, the particle-wise method, relies on a loop over the list containing all particles. To perform the neighbor search, the neighboring data particles present in the surrounded branches, need to be loaded from the main memory. Similarly, the neighboring data particles present in the surrounded branches, need also to be loaded from the main memory. In the worst case, it results that the data have to be accessed from the main memory, instead of cache memory, which is time consuming. Second, the branch-wise method relies on a loop over the incomplete octree branches. Using this method, only one load of the particles data presented in the surrounding branches is required to complete the search process for each particle in a given branch. As the required data are located into the cache, the benefit of this method is that less data have to be loaded from the main memory, which is time saving. Third, the modified branch-wise method relies on the branch-wise method with a modified incomplete octree. According to Monaghan [58], the convenient value for the branch size is $2h$. As this criterion is not always achieved with the branch-wise method, the latter is modified in order to always reach this value. Therefore, the smallest branch size is set to $2h$ and the size of upper branches are set accordingly. The time measurements for the three search methods are given in Figure 2.8. For this purpose, particles were distributed randomly inside a unit cube and then the search time was measured. These measurements were performed on an Intel Xeon-based processor.

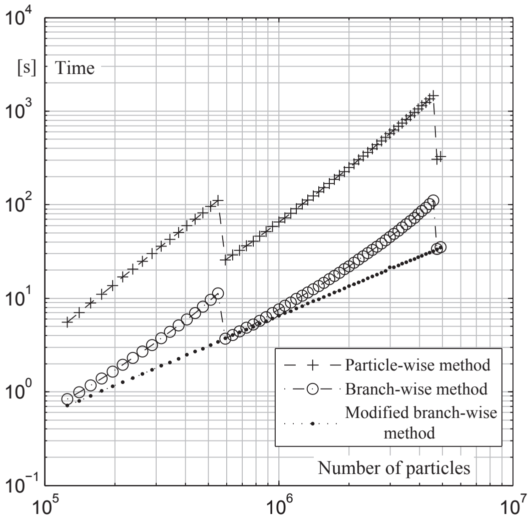
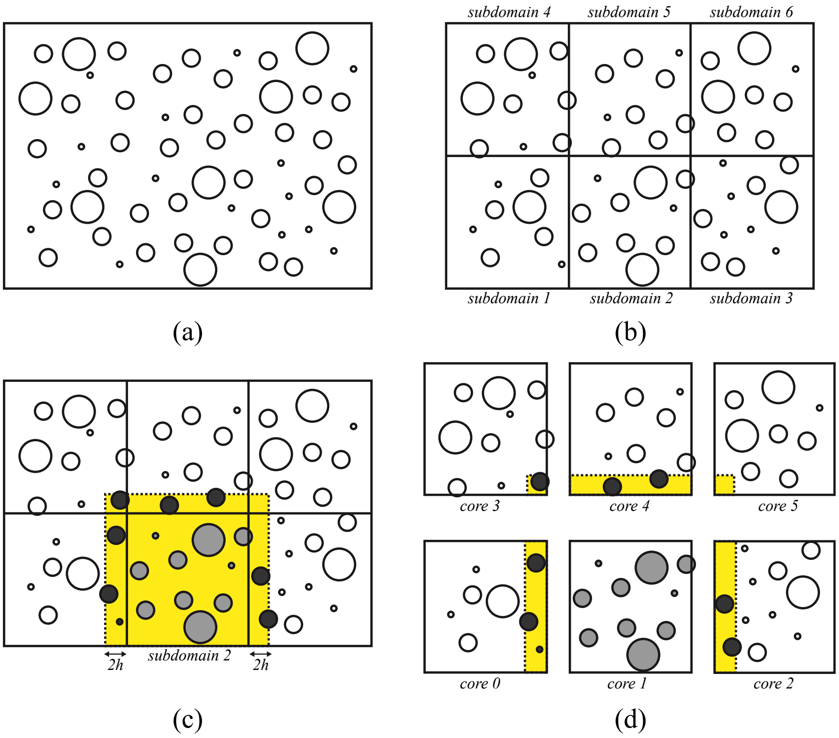


Figure 2.8 – Search time measurements for the three search methods [76].

2.4.3 Parallelization

In the present study, the domain decomposition is integrated in *SPHEROS*. During the initialization process, the whole domain is decomposed into $nd_1 \times nd_2 \times nd_3$ subdomains, as shown in Figure 2.9(b). Each subdomain is then assigned to a specific processor, where the latter can read its own particles position, velocity and density to proceed the computation. Since each subdomain is assigned to a specific processor, linkages between processors are mandatory and each subdomain needs the particle information of its neighboring subdomains. These required data are obtained by expanding the subdomain, followed by the recording of the overlap region on its neighbors. The overlap region is responsible for sending all corresponding data to the appropriate subdomain. This process is illustrated in Figure 2.9(c) and (d). The expansion size determines the amount of data sent to the subdomain of interest i.e. the higher the expansion, the higher the number of particles has to be sent. The subdomain 2 expands and overlaps all the other subdomains, see Figure 2.9(c). Then, all others subdomains send the information about the particles, colored in black, to subdomain 2, as depicted in Figure 2.9(d).



(a) (b)

(c) (d)

Figure 2.9 – Domain decomposition and overlapping subdomains [76].

The MPI library is used in *SPHEROS* to handle inter-processor communications and synchronization. The communication process, based on the aforementioned strategy, has the advantage of having autonomous processors, more suited for massively parallel simulations. Moreover, point-to-point communication with non-blocking *MPI_Isend* and *MPI_Irecv* is used, which enables the code to overlap communication and computation.

As the particles are not fixed in space and could move freely between the subdomains, it is required to establish a particles migration procedure. To handle this type of migration, particles are divided in two groups called "active" and "inactive" groups. The active group contains particles that belong to the subdomain of interest and the inactive group contains the received particles which belong to its neighboring subdomains. At the beginning of each time step and after the communication procedure, both groups of particles are checked in two steps. First, all inactive particles are added to the active group. Second, all active particles, which are out of range of the subdomain of interest, are transferred to the inactive group. To avoid particle accumulation, inactive particles are removed before starting a new communication process.

2.4.4 Solver implementation

The simulation parameters and solvers definition are written in a control text file, which is read by *SPHEROS* at the beginning of the simulation. This file contains the following sections: container, domain, state equation and solvers. The container includes the name of the simulation, the parameters for the time step computation, the export of the results and the log file of the solver. The domain section contains the domain size and decomposition as well as the adaptive domain decomposition parameters. The state equation includes the parameters for the equation of state (2.14). The solver sections contain the definition for each solver e.g. fluid, inlet, wall boundary and sample for the simulation of an impinging jet on a flat plate or Pelton buckets.

After loading all the required parameters for the computation from the control text file, the initial particle location and reference spacing is loaded from HDF5 or text files. Then, the variables data are shared with the neighboring subdomains and a search for the neighboring particles is performed. Next, the solver variables are initialized for the first iteration and the initial state of the computation is saved in the HDF5 results files. Finally, the computation enters the solver iteration presented in Algorithm 1.

Algorithm 1 Description of a FVPM solver iteration for a fluid simulation.

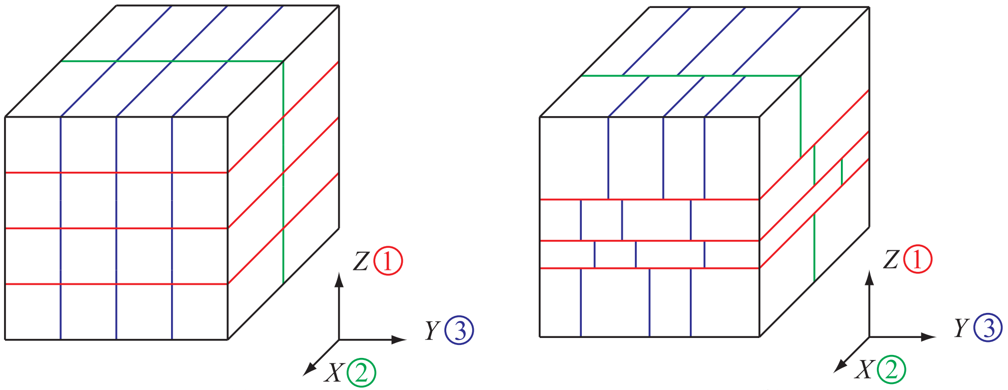
```
for each time step $\Delta t$ do
    share variables data with the neighboring subdomains
    compute $\boldsymbol{\Gamma}_{ij}$
    compute forces $\boldsymbol{f}_i^{(t)} = \sum_j \left(2\mu^* \boldsymbol{D}_{ij}^{(t)} - p_{ij}^{(t)}\mathbf{I}\right) \cdot \boldsymbol{\Delta}_{ij}$
    compute AUSM$^+$
    compute mass change rate $\dot{m}_i^{(t)} = \sum_j \left(aM_{ij}^{(t)} \rho_{ij,\mathrm{AUSM}^+}^{(t)} \boldsymbol{n}_{ij} - \boldsymbol{R}_{ij}^{(t)}\right) \cdot \boldsymbol{\Delta}_{ij}$
    compute volume change rate $\dot{V}_i^{(t)} = \sum_j \left(\dot{\boldsymbol{X}}_j^{(t)} \cdot \boldsymbol{\Gamma}_{ij} - \dot{\boldsymbol{X}}_i^{(t)} \cdot \boldsymbol{\Gamma}_{ji}\right) + \dot{\boldsymbol{X}}_i^{(t)} \cdot \sum_j \boldsymbol{\Delta}_{ij}$
    update momentum $(m_i \boldsymbol{C}_i)^{(t+\frac{\Delta t}{2})} = (m_i \boldsymbol{C}_i)^{(t)} + \boldsymbol{f}_i^{(t)} \frac{\Delta t}{2}$
    update mass $m_i^{(t+\frac{\Delta t}{2})} = m_i^{(t)} + \dot{m}_i^{(t)} \frac{\Delta t}{2}$
    update volume $V_i^{(t+\frac{\Delta t}{2})} = V_i^{(t)} + \dot{V}_i^{(t)} \frac{\Delta t}{2}$
    update density $\rho_i = m_i / V_i$ and compute $p_i$ from equation of state
    update particle velocity $\dot{\boldsymbol{X}}_i^{(t+\frac{\Delta t}{2})} = \boldsymbol{C}_i^{(t+\frac{\Delta t}{2})} + \dot{\boldsymbol{X}}_i'$
    update position $\boldsymbol{X}_i^{(t+\Delta t)} = \boldsymbol{X}_i^{(t)} + \dot{\boldsymbol{X}}_i^{(t+\Delta t)} \Delta t$
    enforce displacement of the boundary conditions
    share variables data with the neighboring subdomains
    search neighboring particles
      if (ADD is required)
        then    enter ADD process and search neighboring particles
    compute $\boldsymbol{\Gamma}_{ij}$
    compute forces $\boldsymbol{f}_i^{(t+\frac{\Delta t}{2})}$
    compute AUSM$^+$
    compute mass change rate $\dot{m}_i^{(t+\frac{\Delta t}{2})}$
    compute volume change rate $\dot{V}_i^{(t+\frac{\Delta t}{2})}$
    update momentum $(m_i \boldsymbol{C}_i)^{(t+\Delta t)} = (m_i \boldsymbol{C}_i)^{(t)} + \boldsymbol{f}_i^{(t+\frac{\Delta t}{2})} \Delta t$
    update mass $m_i^{(t+\Delta t)} = m_i^{(t)} + \dot{m}_i^{(t+\frac{\Delta t}{2})} \Delta t$
    update volume $V_i^{(t+\Delta t)} = V_i^{(t)} + \dot{V}_i^{(t+\frac{\Delta t}{2})} \Delta t$
    update density $\rho_i = m_i / V_i$ and compute $p_i$ from equation of state
    update particle velocity $\dot{\boldsymbol{X}}_i^{(t+\Delta t)}$
    $t \leftarrow t + \Delta t$
    solver iteration output
      if (export results is required)
        then    HDF5 output
    compute new $\Delta t$
end for
```

3 Adaptive Domain Decomposition

3.1 Recursive multi-section algorithm

The Adaptive Domain Decomposition (ADD) is a useful tool, which enables to fully exploit the computational power of a cluster by adapting the subdomains size to ensure a uniform particles load. In particle-based simulations, the load balancing process is usually done by changing the domains size according to one given direction, e.g. [74] and [16]. In the literature, two main strategies have been used for 3-D adaptive process of particle-based simulations. The first one is based on the space filling curves used by Springel [73], which map the 3-D space into a 1-D curve. The second one is the recursive multi-section algorithm of Ishiyama et al. [29]. In the present study, the recursive multi-section algorithm is used to partition the computational domain but the adaptive process is computed by a new approach based on the real subdomains load instead of using the sampling approach from Guibert et al. [23].

The recursive multi-section algorithm of Ishiyama et al. [29] decomposes the 3-D computational domain into rectangular subdomains. The recursive multi-section decomposition performed by this algorithm is presented in Figure 3.1(a) for a given $nd_1 \times nd_2 \times nd_3$ uniform decomposition. This decomposition depends on the user-prescribed main directions (1, 2 and 3 instead of X, Y and Z). The recursive multi-section algorithm first splits the computational domain in nd_1 blocks according to the first direction. Then, each nd_1 new formed block of subdomains are split, independently from each other, in nd_2 blocks according to the second direction. Finally, the $nd_1 \times nd_2$ blocks of subdomains are split independently in nd_3 subdomains according to the third direction. An example of adapted domain decomposition using the recursive multi-section algorithm is shown in Figure 3.1(b). The number of block per main direction is given by $\boldsymbol{nB} = \{nd_1, nd_1 \cdot nd_2, nd_1 \cdot nd_2 \cdot nd_3\}$ and the number of subdomains per block is $\boldsymbol{nD} = \{nd_2 \cdot nd_3, nd_3, 1\}$. Consequently, the number of interface to move per main direction is given by $\boldsymbol{nS} = \{nd_1 - 1, nd_1 (nd_2 - 1), nd_1 \cdot nd_2 (nd_3 - 1)\}$.



(a) (b)

Figure 3.1 – Uniform domain decomposition (a) and adapted domain decomposition using the recursive multi-section algorithm (b).

3.2 Adaptive process

During the ADD process, the interfaces between the subdomains are moved to make uniform the load in the subdomains blocks. In the adaptive process of Guibert et al. [23], the displacement of the subdomains interfaces are computed by a sampling, which probes the particles number in the computational domain. Although the sampling approximation allows a fast estimation of the load in the subdomains, a particular attention has to be paid in the case of highly dispersed flows or highly parallelized simulations. Therefore, in the proposed adaptive process, the displacements are computed from the real load in the subdomains.

The maximum load in the subdomains blocks $\boldsymbol{L}_{\max}$ is computed at each iteration of the solver and for each main direction. If one or several components of the maximum load are greater than a user-prescribed load limit L_{lim}, the solver enters the adaptive process according the first main direction involved. When entering the adaptive process, the solver communicates the particles information from the neighboring subdomains according to the $2h$ overlap region. Therefore, each interface between the subdomains blocks can be moved of a maximum distance $m_{\max} = 2h$ to ensure that the ADD process do not lose particles. Moreover, the minimum subdomains size is set by $d_{\min} = 6h$ to avoid inefficient computation by increasing drastically the size of the inactive group of particles.

The adaptive process is applied to each group of subdomains blocks for the main direction of interest. A group is composed of all the blocks included in the same block at the previous main direction. An example of a group is shown in Figure 3.2 for the third main direction. According to this example, the differences between the blocks load and the ideal load are computed for each interface by

$$\begin{cases} L_1 = I - N_A \\ L_2 = L_1 + I - N_B \\ L_3 = L_2 + I - N_C \end{cases} \tag{3.1}$$

where $I = (N_A + N_B + N_C + N_D) / nd_3$ is the ideal load per group. The direction of the interfaces displacements is imposed by the sign of L. In the case where $L = 0$, the interface is not moved during the adaptive process.

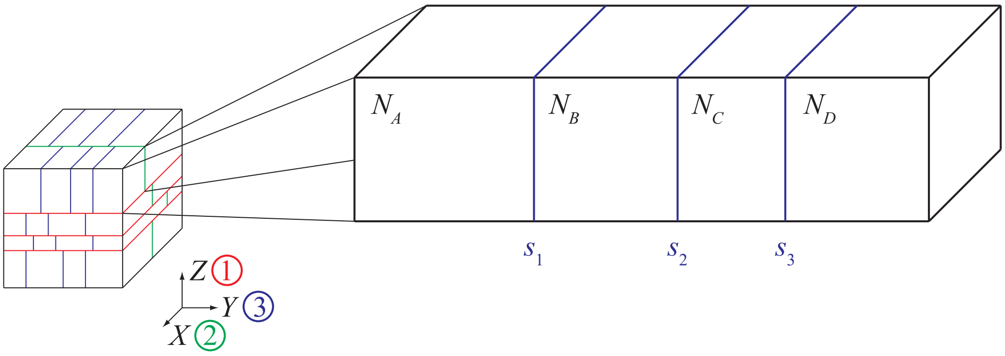


Figure 3.2 – Example of a subdomains group for the third main direction. The number of particles in the subdomains are N_A, N_B, N_C and N_D. The interfaces s_1, s_2 and s_3 are moved by the adaptive process.

In order to compute the magnitude of the interface displacement, the gradient of load is evaluated and stored for each interface. This gradient of load is approximated from an imposed displacement of $m_{\max}$, and so, is expressed as

$$G = \frac{\Delta N}{m_{\max}} \tag{3.2}$$

where ΔN represents the difference of particles numbers that results from the $m_{\max}$ displacement of the interface. As the particles data of the overlapping zone are included in the subdomains data, no extra particles search is required to evaluate this gradient. In the case where the gradient of load is equal to zero for an interface, the magnitude of the displacement is set to $m_{\max}$.

Knowing the load differences from (3.1) and the approximated gradients from (3.2), the magnitudes of the interfaces displacements are expressed as

$$\left\{ \begin{array}{l} m_1 = L_1 / G_1 \\ m_2 = L_2 / G_2 \\ m_3 = L_3 / G_3 \end{array} \right. . \tag{3.3}$$

After computing the displacements in the blocks using (3.3), their magnitudes are limited to $m_{\max}$. This limitation is applied to ensure that the interface displacement remains in the overlapping region of the subdomains, which contains the neighboring informations required to transfer the particles data. An example of interfaces displacements in a group is presented in Figure 3.3 for the third main direction.

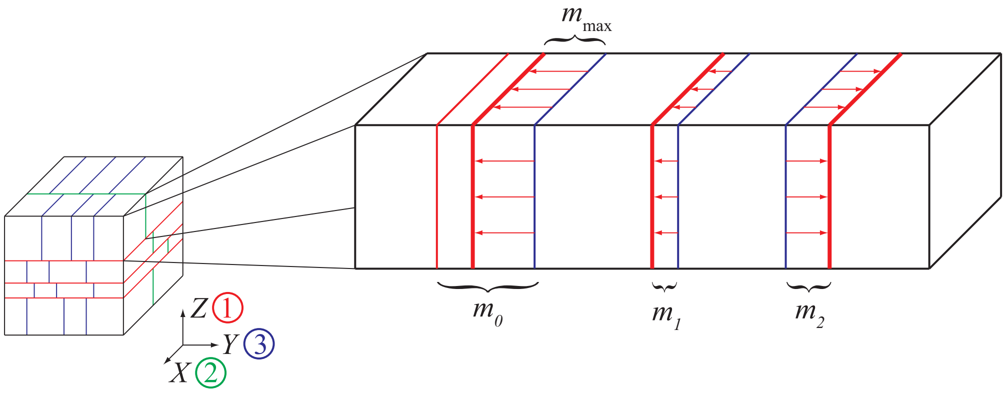


Figure 3.3 – Example of the interfaces displacements in a subdomains group for the third main direction. The displacements of the interfaces are given by m and $m_{\max}$ denotes the maximum displacement allowed.

To improve the adaptive process, a special treatment is applied to the interfaces which are linked to an empty subdomains block. When a subdomain block is empty, the displacement is allowed to be greater than $m_{\max}$. In the case were both blocks are empty, the displacement of the interface is computed according to $d_{\min}$. In the case were only one block is empty, the displacement is computed according to d_f the free distance between the interface and the particles. In the latter case, the displacement is computed by

$$m = \max\left(d_f - h, m_{\max}\right). \tag{3.4}$$

At the end of the adaptive process, the size of subdomains are checked to ensure that the computed displacements will not generate subdomains size smaller than $d_{\min}$. Then, a communicate and search process is required to update the active and inactive groups in the subdomains. Moreover, two additional checks are performed to avoid that the adaptive process gets stuck. The first check reports if the differences between the computed displacements for the three last iterations are within 30% for all the interfaces of the main direction. The second check reports if the number of consecutive iterations for the main direction of interest are greater than a user-specified value. If at least one of these checks are validated, the adaptive process can not enter this main direction for a given value of solver iterations.

3.3 Validation

3.3.1 Orbital shaking

The validation of the proposed ADD process is performed on the orbital shaking test case, described in Figure 3.4. The orbital shaking test case imposes an orbital motion to a cylinder partially filled with water. The orbital shaking process is often used in biomedical applications for the growth of cell-cultures in bioreactors. The cylinder's orientation is fixed in the Cartesian coordinate system and a circular motion is imposed to its center around the Z axis. The diameter of the cylinder is $D = 144$ mm and the initial water height is $Z_0 = 75$ mm. The rotational speed of the boundary is set to 85 rpm and the shaking diameter is $d_s = 11.4$ mm, which corresponds to one of the operating conditions investigated by Reclari [71].

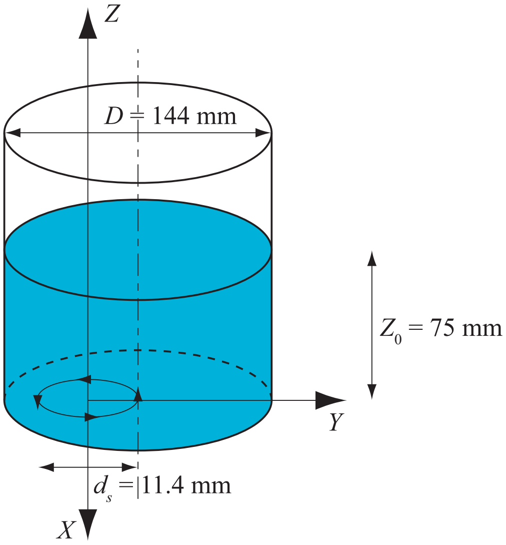


Figure 3.4 – Description of the orbital shaking test case. The diameter of the cylinder is D, the initial water height is Z_0 and d_s is the shaking diameter.

The case study is simulated with the FPM solver of *SPHEROS*. The interest of the FPM solver is its ease to capture accurately the waves of the free surface compared to grid-based code. An example of FPM result is represented in Figure 3.5 using the *POV-Ray* software to render the free surface.

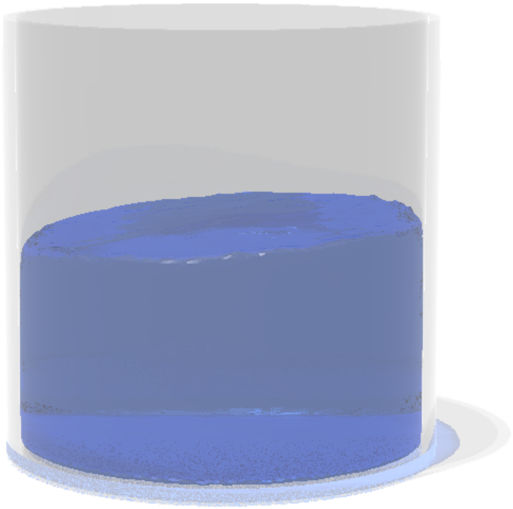


Figure 3.5 – Visualization of the *SPHEROS* result using the *POV-Ray* rendering software.

The evolution of the free surface on the cylinder's boundary is presented in Figure 3.6 for a spatial discretization of $Z_0/X_{\text{ref}} = 40$. The FPM result is first run during 40 cylinder's revolutions to obtain a converged flow pattern. Then, 20 revolutions are used to generate the FPM evolution of the free surface given in Figure 3.6. The FPM result is compared to measurements performed by Reclari [71]. Despite a larger standard deviation, the FPM result is in good agreement with the experimental data.

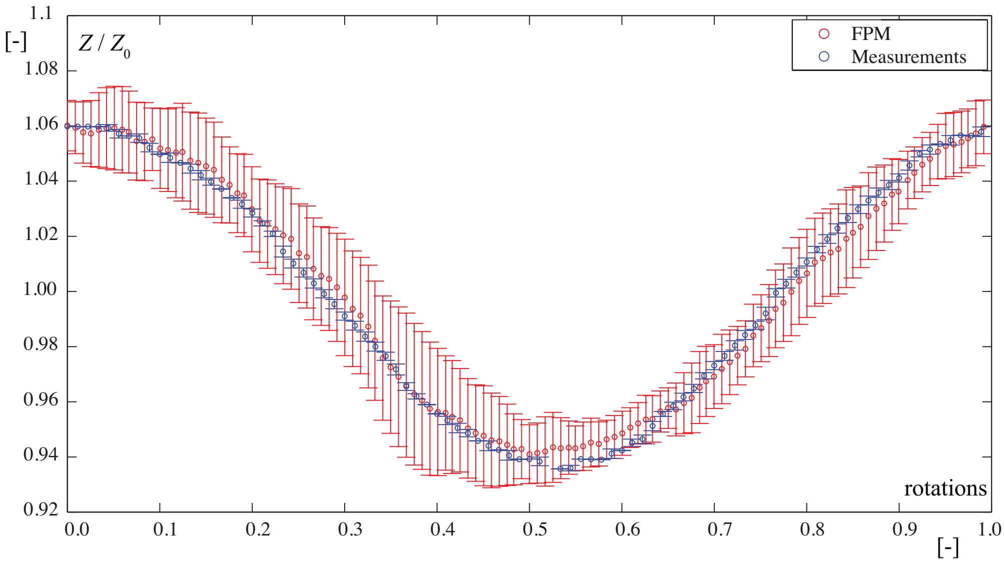
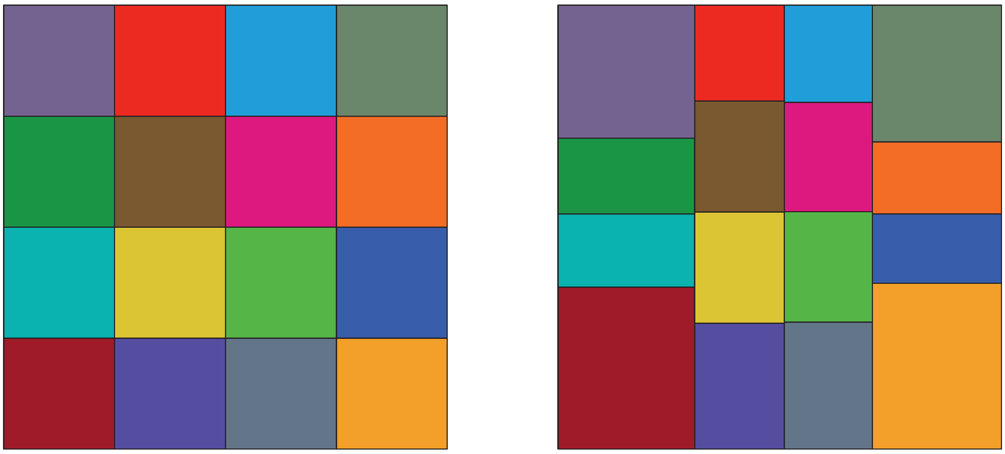


Figure 3.6 – Evolution of the free surface on the cylinder's boundary according to the revolutions. Comparison between the measurements from Reclari [71] and the FPM result for a spatial discretization of $Z_0/X_{\text{ref}} = 40$.

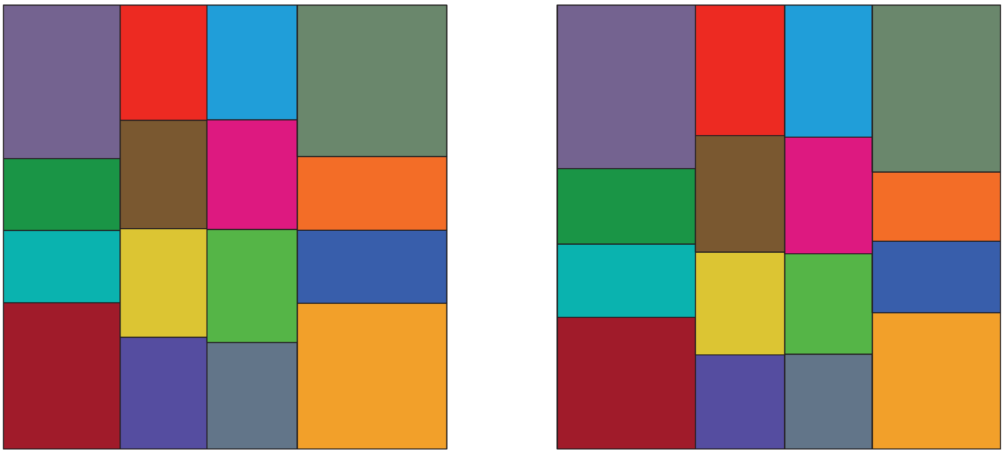
In order to validate the ADD process, the orbital shaking is simulated with the spatial discretization of $Z_0/X_{\text{ref}} = 40$ on 16 cores starting from a uniform domain decomposition $nd_1 = 4$, $nd_2 = 4$ and $nd_3 = 1$. The main directions correspond to the cartesian coordinate system. The load limit $L_{\text{lim}} = 5\%$ is chosen for the ADD process. The Figure 3.7 shows the evolution of the subdomains size according to the position of the cylinder. Thanks to the ADD process, the motion of the cylinder is followed by looking at the size of the subdomains in Figure 3.7. This figure also reports the evolution of L_{max} the maximum load in the subdomains. In this case, the maximum load is given by $L_{\text{max}} = L_{\text{max},2}$. The initial uniform domain decomposition features a maximum load in the subdomains 8.44 times greater than the chosen load limit. Then, L_{max} remains under L_{lim} for all the other cylinder's position presented in Figure 3.7.

The evolution of the maximum load is presented in Figure 3.8 as a function of the simulated time. After only three iterations of the ADD process, the maximum load becomes lower than L_{lim}. During the simulation, L_{max} increases due to the rotation of the cylinder. When L_{max} is greater than L_{lim}, the ADD is activated to reduce the maximum load in the subdomains. The Figure 3.8 highlights the efficiency of the adaptive process applied to the orbital shaking test case.



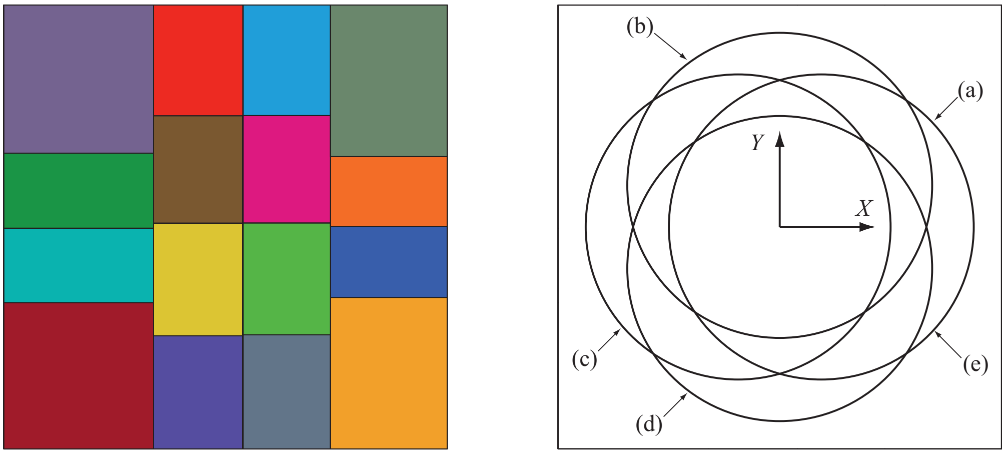
(a) 0 Rotation, $L_{max} / L_{lim} = 8.44$

(b) 0.25 Rotations, $L_{max} / L_{lim} = 0.99$



(c) 0.5 Rotations, $L_{max} / L_{lim} = 0.27$

(d) 0.75 Rotations, $L_{max} / L_{lim} = 0.29$



(e) 1 Rotation, $L_{max} / L_{lim} = 0.53$

(f) Position of the cylinder

Figure 3.7 – Size of the subdomains according to the cylinder's rotations. L_{max} denotes the maximum load in the subdomains and L_{lim} is the imposed load limit for the ADD process.

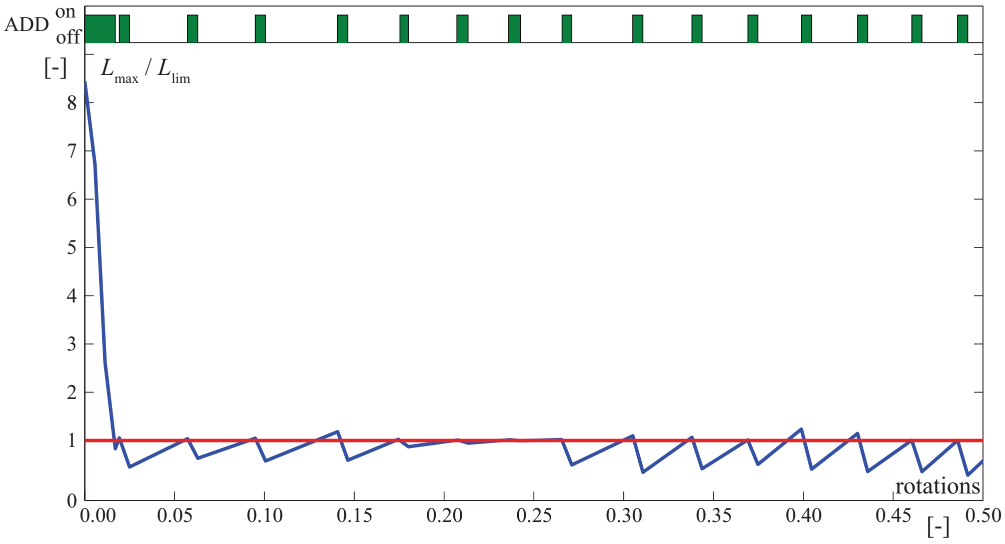


Figure 3.8 – Evolution of the maximum load in function of the cylinder's rotations. L_{max} denotes the maximum load in the subdomains and L_{lim} is the imposed load limit for the ADD process.

The efficiency of the proposed ADD process, compared to a fixed uniform domain decomposition, is validated by a strong scaling analysis of the orbital shaking test case. The strong scaling reports the computing time with respect to the number of cores, keeping the total load constant. The strong scaling is performed on two different clusters: an IBM blade center and the *Lemanicus* BG/Q.

First, the computations are run on an IBM blade center, which consists of 6 blades with 32 cores E5 2670 at 2.6 GHz (hyper threading 2x16) and 32 Gb of memory. The simulated time corresponds to 1.25 rotations of the cylinder. Two different discretizations $Z_0 / X_{\text{ref}} = 40$ and $Z_0 / X_{\text{ref}} = 80$ are investigated, which feature 256'671 and 782'157 particles respectively. The computations are run with 2, 4, 8, 16, 32, 64 and 128 cores. The strong scaling analysis given in Figure 3.9, compares the speed up for the simulation with ADD to the same simulation with a fixed Uniform Domain Decomposition (UDD). The measured time for the simulations on 2 cores are chosen as the reference for the speed up curves. Up to 4 cores, there is no improvement between the ADD and UDD because the uniform decomposition corresponds approximately to the symmetry axis of the computational domain. Between 4 and 32 cores, the uniform decomposition shows a stagnation in the computational time. This stagnation is explained by the lost of efficiency due to the unbalanced load. The ADD presents a satisfactory slope up to 16 cores, which highlight the efficiency of this strategy. The change of slope after 16 cores is due to the rather slow blade to blade connection (Ethernet 1 Gb cable). The simulations with the uniform domain decomposition has a shifted behavior after 32 cores compared to the ADD simulations. This is explained by the higher number of decomposition in the X and Y axis, which leads to smaller subdomains sizes.

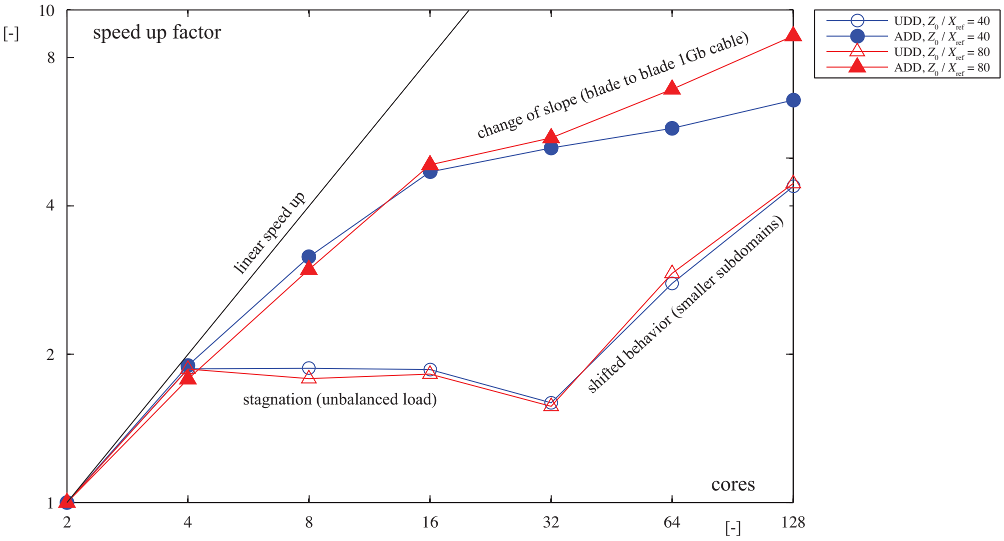


Figure 3.9 – Strong scaling analysis of the orbital shaking test case for two different spatial discretizations. The simulations are performed with a fixed uniform domain decomposition (UDD) or with the adaptive domain decomposition process (ADD).

Second, the computations are run on the *Lemanicus* BG/Q, which features one rack of 1’024 sixteen-cores computing nodes PowerA2 at 1.6 GHz, 16 GB of memory per computing node and a 5-D torus communication network. The orbital shaking is simulated using the FVPM solver and a spatial discretization of $Z_0/X_{\text{ref}} = 85$, which leads to 1’927’260 particles. The strong scaling is presented in Figure 3.10 for both ADD and UDD between 2 and 16’384 cores.

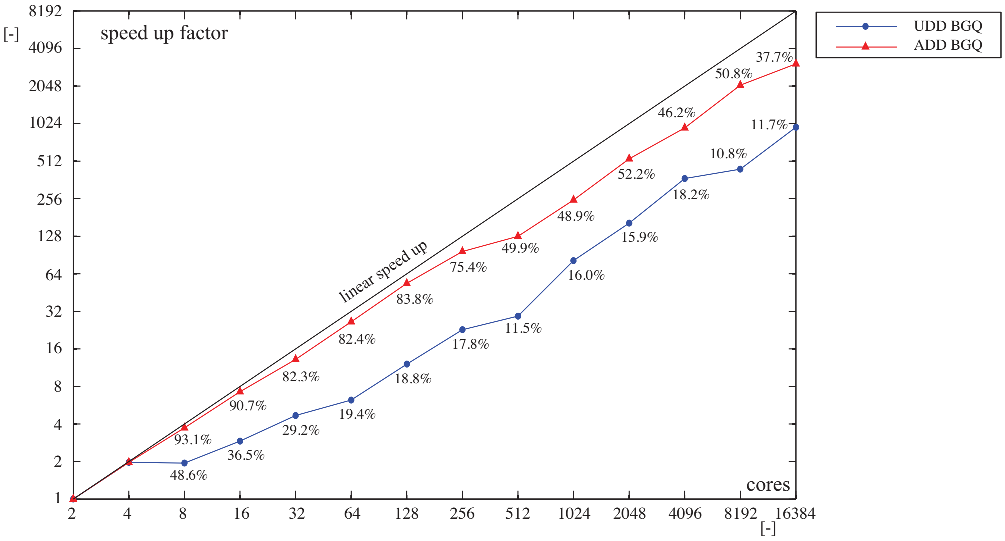


Figure 3.10 – Strong scaling analysis of the orbital shaking test case on the *Lemanicus* BG/Q. The simulations are performed with a fixed uniform domain decomposition (UDD) or with the adaptive domain decomposition process (ADD).

In comparison with Figure 3.9, the *Lemanicus* BG/Q is more efficient for highly-parallelized computations due to a better communication strategy. Moreover, the Figure 3.10 highlights the efficiency improvement of the ADD compared to UDD. Indeed, the adaptive process allows a computation 3 times faster in average for this case study. However, the decrease of efficiency for ADD at 16'384 cores highlights the limitation of the recursive multi-section algorithm. In the latter simulation, the ideal load per core is less than 120 particles. Therefore, the adaptive process is not able to decrease the maximum load under L_{lim} because the subdomains size is limited by d_{min}.

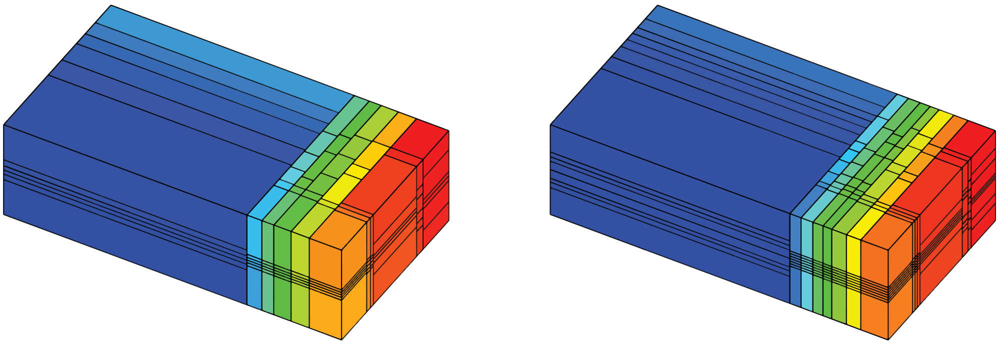
3.3.2 Pelton buckets

The validation of the ADD process is performed on the rotating Pelton buckets test case, which features a high-speed water jet impinging on three rotating buckets. This test case is simulated with the FVPM solver of *SPHEROS* on the *Lemanicus* BG/Q. The detailed case study, convergence study and results are presented in Chapter 5. The present study reports the influence of the domain decomposition and adaptive process on the subdomains load and computing time. The influence of the domain decompositions 6 × 6 × 6, 8 × 8 × 8, 32 × 6 × 6 and 18 × 8 × 8 on the subdomains sizes is highlighted in Figures 3.11(a), (b), (c) and (d). In these figures, the subdomains sizes are obtained from the converged ADD process on the initial particles position, which is given in Figure 3.11(e). The maximum load in the main directions and mean particles number per core are given in Table 3.1 for all the domain decompositions investigated.

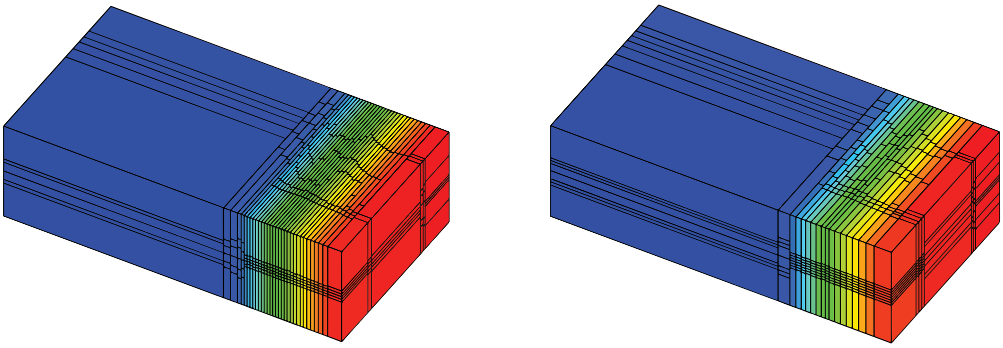
Table 3.1 – Influence of the domain decomposition on the maximum load in the main directions and mean particles number per core.

Decomposition	$\boldsymbol{L_{\text{max},1}/L_{\text{lim}}}$	$\boldsymbol{L_{\text{max},2}/L_{\text{lim}}}$	$\boldsymbol{L_{\text{max},3}/L_{\text{lim}}}$	$\boldsymbol{N_{\text{mean}}}$
6 × 6 × 6	0.77	0.97	0.96	1'856
8 × 8 × 8	0.74	0.65	4.28	783
32 × 6 × 6	0.73	0.91	4.55	348
18 × 8 × 8	0.72	0.93	7.87	348

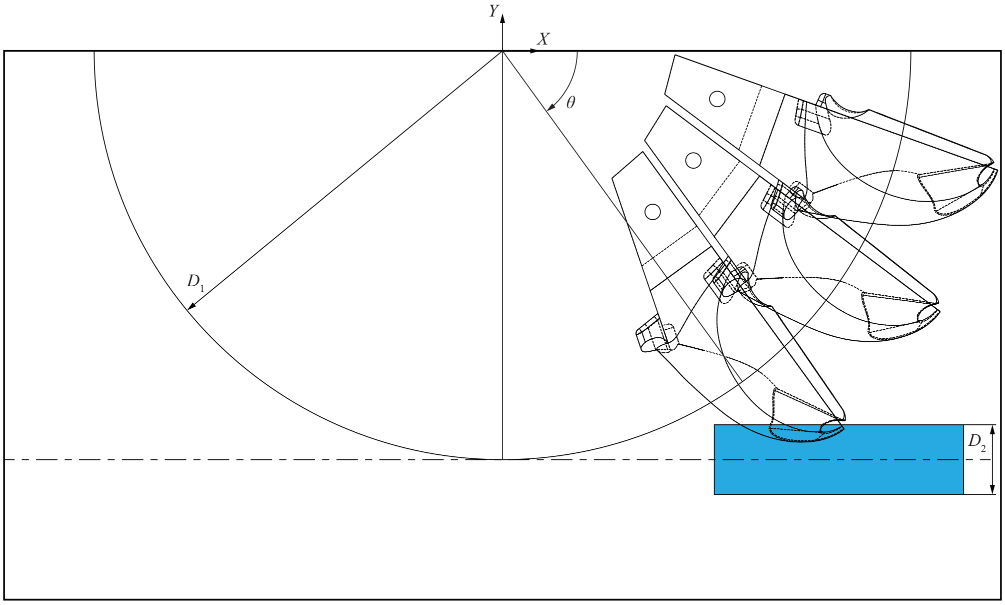
According to this table, the converged ADD process for the 6 × 6 × 6 domain decomposition is able to reach L_{lim} for all the main directions. However, for the 8 × 8 × 8 decomposition, the ADD process is not able to reduce $L_{\text{max},2}$ under L_{lim} due to the d_{min} subdomains size limitation. To perform efficient simulations on the *Lemanicus* BG/Q, a mean particles number per core between 300 and 400 is aimed. Different domain decompositions may be used to reach this goal. The comparison between the 32 × 6 × 6 and the 18 × 8 × 8 decompositions highlights that the first one leads to a smaller $\boldsymbol{L}_{\text{max}}$. Therefore, the 32 × 6 × 6 decomposition is chosen for the following analysis despite L_{lim} is not reached for $L_{\text{max},2}$.



(a) Domain decomposition: 6×6×6 (b) Domain decomposition: 8×8×8



(c) Domain decomposition: 32×6×6 (d) Domain decomposition: 18×8×8



(e) Initial position of the three buckets and high-speed water jet

Figure 3.11 – Influence of the domain decomposition on the subdomains sizes. The subdomains sizes are obtained from the converged ADD process on the initial particles position.

The evolution of the maximum load is presented in Figure 3.12 in function of the buckets position. The maximum load in the first direction reaches the fixed load limit of 20% during the simulation. However, the maximum load in the second and third main directions are not able to reach this limit due to the d_{min} limitation. As expected from the results in Table 3.1, the third main direction presents the highest maximum load during the simulation. The maximum loads evolve with the buckets position due to the rotation of the geometry and evolution of the flow pattern.

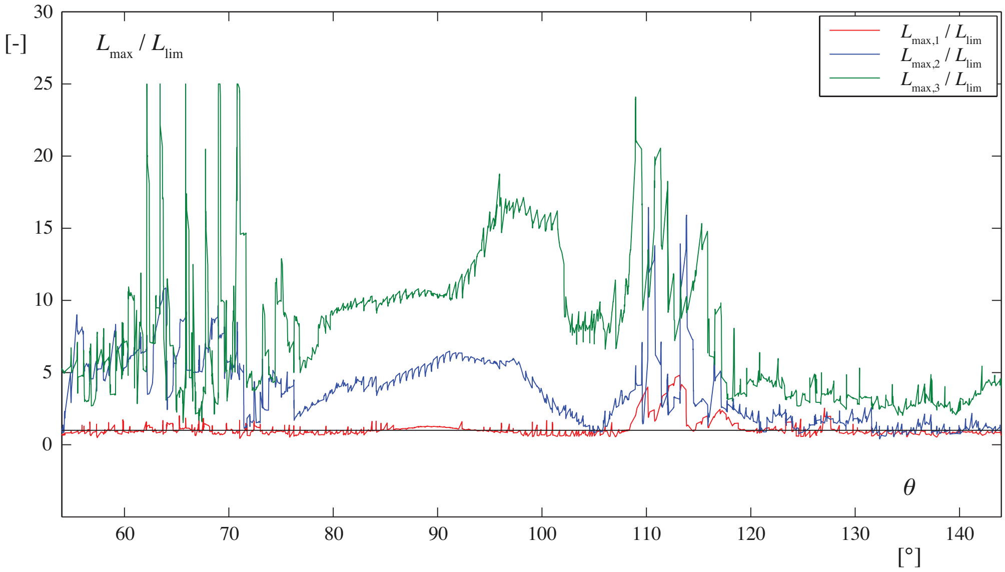


Figure 3.12 – Evolution of the maximum load in function of the buckets position. $L_{\text{max},i}$ denotes the maximum load in the subdomains according to the main direction i and L_{lim} is the imposed load limit for the ADD process.

The evolution of the computing time for a solver iteration is reported in Figure 3.13 in function of the buckets position for the simulations with ADD and UDD. This computing time is normalized by the mean ADD time per iteration. In comparison to ADD, the UDD time increases significantly during the computations, which highlights the efficiency of the adaptive process. The computing time for the ADD is less than one tenth of the time for a solver iteration. Therefore, the cost of the ADD for a computation is not significant. Moreover, the number of ADD iterations divided by the number of solver iterations is equal to 0.2658 for this test case. The maximum number of consecutive ADD iterations according to a main direction is set to 10 and the number of solver iterations to wait when the ADD get stuck is set to 100. The total computing time for the simulation with ADD is 75 hours and this time increases to 254 hours for the simulations with UDD. Consequently, the ADD process decreases the computing time by a factor 3.4, which corresponds to the value obtained from the strong scaling analysis of the orbital shaking test case.

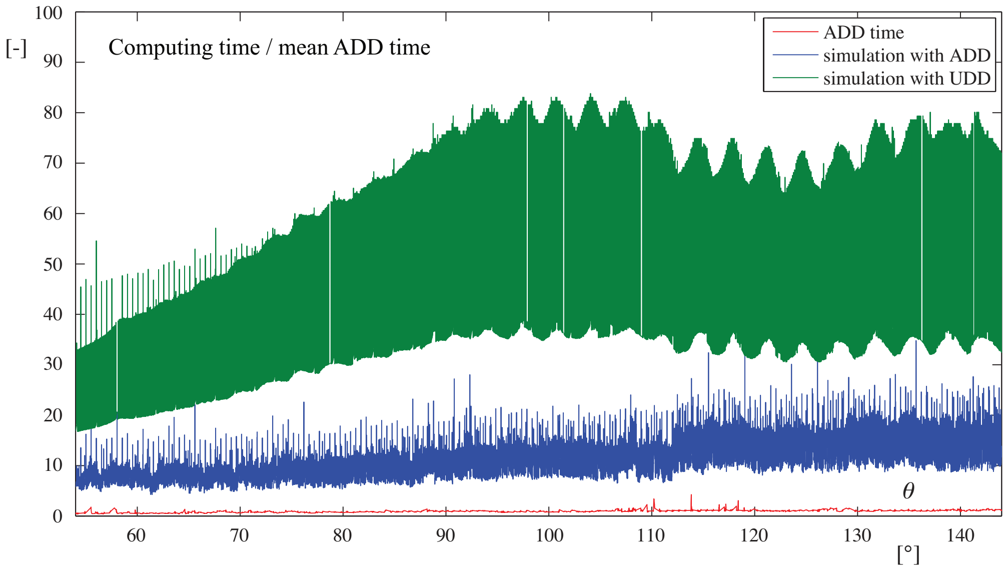


Figure 3.13 – Computing time for a solver iteration in function of the buckets position. Comparison between simulations with ADD and UDD.

4 Impinging Jet on a Flat Plate

4.1 Case study

The purpose of this chapter is to validate that the deviation of the high-speed water jet is captured properly by the FPM simulations. Therefore, a simplified case study is investigated, which consists of a jet impinging a flat plate at three different angles. The FPM simulations are done with the software *SPHEROS*, developed at EPFL since 2010 for the purpose of efficient massively parallel simulations [32]. The FPM simulations are validated with grid-based numerical simulations and experimental data obtained by Kvicinsky et al. [41]. The grid-based numerical simulations were performed with the commercial software *ANSYS-CFX* by using Volume of Fluid (VOF) two-phase method.

An outline of the experimental setup is given in Figure 4.1. It consists of a flat plate, which can tilt around the Y axis of the Cartesian coordinate system. A variable stroke needle controls the diameter of the high-speed jet generated at the injector nozzle. The inlet section is located at a length $L = 0.1\,\mathrm{m}$ above the plate center, where the jet is stabilized with a diameter $D = 0.03\,\mathrm{m}$ and a discharge velocity $C_{\mathrm{max}} = 19.8\,\mathrm{m\,s^{-1}}$. At the inlet section, the velocity profile is imposed according to the measurements of Kvicinsky et al. [41] for taking into account the perturbation of the flow by the injector needle, see Figure 4.2.

The experimental data consist of pressure measurements on the flat plate and location of the free surface for three different impinging angles: $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. Pictures of the experimental setup are given in Figure 4.3. In this Chapter the FPM simulations of the impinging jet using *SPHEROS* are compared to the CFD and measurements from Kvicinsky et al. [41]. The free surface location as well as the pressure coefficient

$$C_p = \frac{p - p_{\mathrm{ref}}}{\frac{1}{2}\rho_{\mathrm{ref}} C_{\mathrm{max}}^2} \tag{4.1}$$

are used to quantitatively compare the results. All the simulations are run on a Intel Xeon CPU E5 2670 at 2.6 GHz with 16 cores and 32 Gb of memory. The domain decomposition is uniform, 4 × 4 × 1 in the X, Y and Z directions.

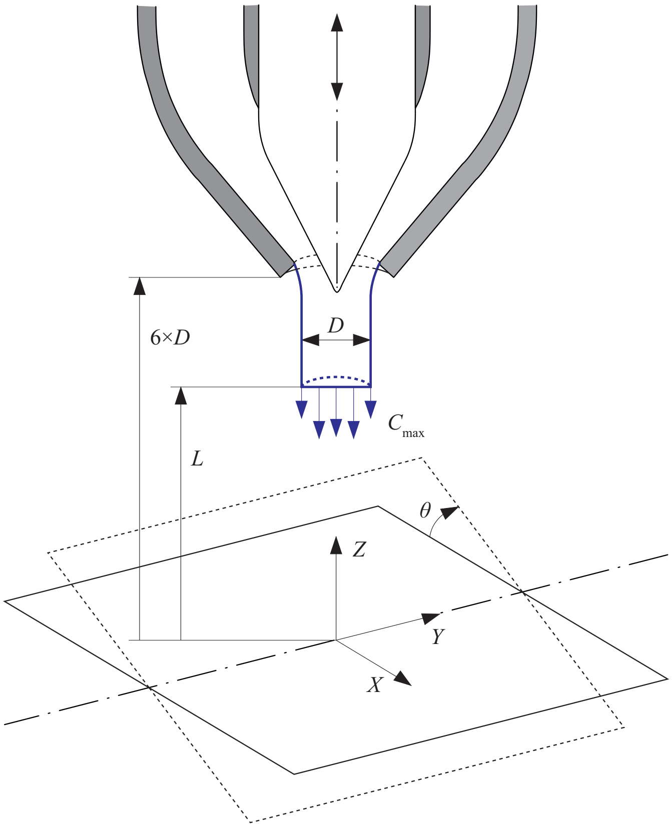


Figure 4.1 – Outline of the impinging jet test case. The Cartesian coordinate system (X, Y and Z) is located on the center of the flat plate. This flat plate can tilt of an angle θ around the Y axis. The inlet of the jet is located at L, D is the jet diameter and $C_{\max}$ is the discharge velocity at the inlet.

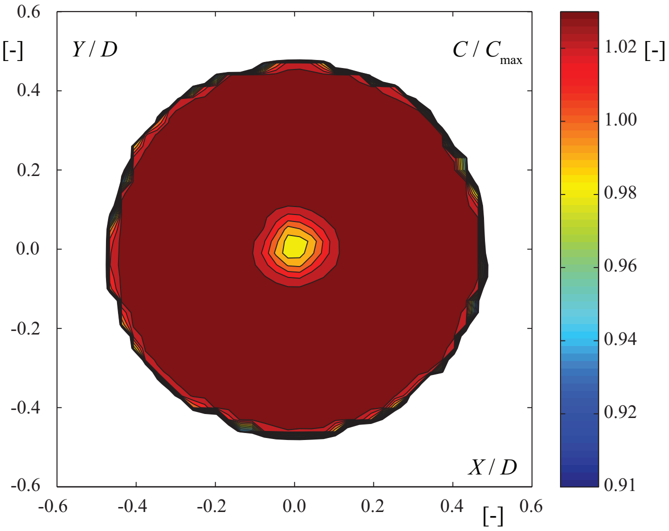
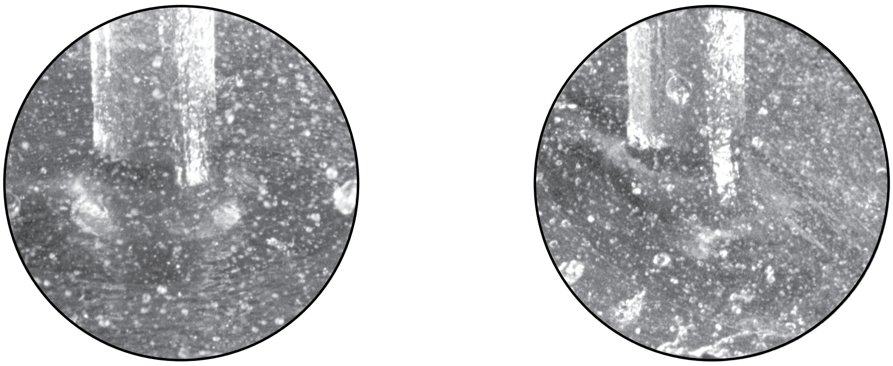
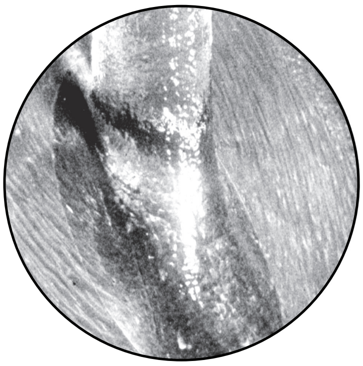


Figure 4.2 – Measured velocity profile on the inlet section [41]. The absolute velocity is C, D is the jet diameter and $C_{\max}$ is the discharge velocity at the inlet.



(a) impinging angle of 0° (b) impinging angle of 30°



(c) impinging angle of 60°

Figure 4.3 – Water jet impinging on the plate at $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$ [38].

4.2 FPM validation

4.2.1 Shifting strategy

The mean C_p profile along the X axis of the flat plate is presented in Figure 4.4 for different shifting strategies. First, the pressure peak experiences instabilities when no shifting is applied, which highlights the importance of the shifting strategies. Second, the simulation with the shifting vector (2.41) from Xu et al. [83] presents a significant increase of the jet diameter due to the artificial spread of particles through the free surface. This behavior is expected because this shifting method was developed in the case of confined flows. The simulation with the shifting vector (2.43) from Jahanbakhsh et al. [31] improves the results. However, the pressure peak is overestimated compared the measurements. Finally, the simulation with the proposed shifting vector (2.46) improves the accuracy of the FPM simulation. The efficiency of this method to avoid the artificial spread of particle through the free surface is highlighted. Moreover, the pressure profile of the simulation with the proposed shifting fits extremely well both the measurements and the CFD results.

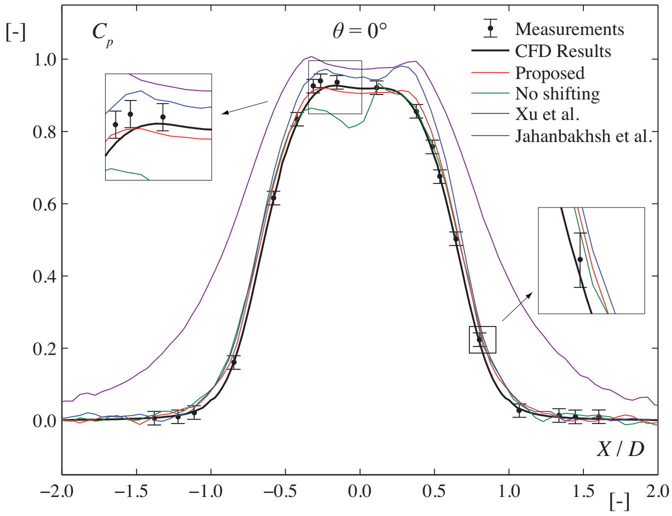


Figure 4.4 – Influence of the particle shifting method on the averaged C_p along the X axis of the flat plate. The pressure peak experiences instabilities when no shifting is applied. The artificial spread of particles through the free surface is highlighted with the shifting vector from Xu et al. [83]. The proposed shifting methods improves the accuracy of the simulation to capture the pressure peak compared to the simulation with the shifting vector from Jahanbakhsh et al. [31].

4.2.2 Inlet conditions

The influence of the imposed velocity profile at inlet is highlighted in Figure 4.5, which represents the mean C_p profile along the X axis of the flat plate for different inlet velocity profiles. The uniform velocity profile does not take into account the disturbance of the flow by the injector's needle, and so, produces a standard bell-shaped pressure profile. The imposed velocity profile from Kvicinsky et al. [41] applies the measured velocity profile from the experiment to the inlet of the computational domain. This imposed profile was also applied to the CFD results. Therefore, the FVPM result with the imposed profile fits better the CFD results and measurements.

At the beginning of the simulation, the computational domain is empty and the water jet is injected from the inlet. Therefore, the impingement starts with a transient behavior and then converges to a steady state. A damping zone is used to reduce the pressure waves, which are reflected from the plate back to the inlet at the beginning of the impingement. In this damping zone, the governing equations follow (2.54) and (2.55) to damp the flow according to its reference state. As the $\theta = 0°$ impinging angle case presents the most reflections of pressure waves, the time history of C_p at the center of the flat plate is shown in Figure 4.6 for this worst case. This time history confirms that the pressure oscillations are stabilized quicker with the damping zone. In the present analysis, the mean C_p profile along the X axis of the flat plate is used to compare the results to the CFD and measurements. Thus, the mean value is performed for a time $3 \le t\, C_{\max}/L \le 6$ to ensure that the simulations are stabilized.

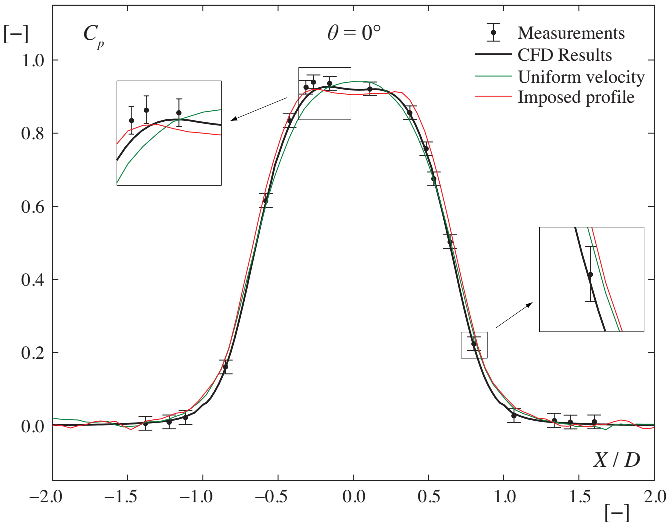


Figure 4.5 – Influence of the inlet velocity profile on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$. The uniform velocity profile generates a bell-shaped pressure profile because it does not take into account the disturbance of the flow by the injector. The imposed profile from Kvicinsky et al. [41] fits properly the CFD results and measurements.

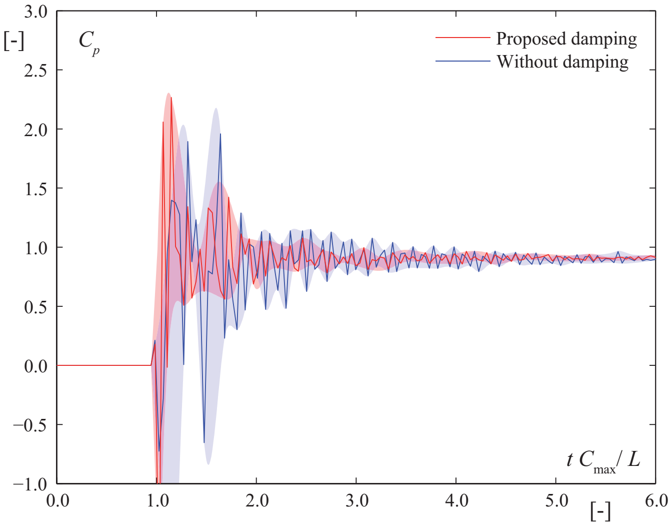


Figure 4.6 – Influence of the damping on the time history of C_p at the center of the flat plate. The pressure waves, which are reflected from the plate back to the inlet, are stabilized quicker with the damping zone.

4.2.3 Artificial viscosity

The influence of the artificial viscosity on the averaged C_p along the X axis of the flat plate is presented in Figure 4.7 for an impinging angle of $\theta = 0°$. The difference between the simulation without artificial viscosity, i.e. $\mu^* = 0$ Pa·s, and the simulation with $\mu^* = 0.001$ Pa·s is negligible, which validates the inviscid flow hypothesis. Indeed, the jet deviation depends on the inertial forces not on the viscous ones. However, in the case where the viscous forces are significant, i.e. $\mu^* = 1$ Pa·s, the viscous forces damp the inlet velocity profile, which produces a standard bell-shaped pressure profile on the flat plate.

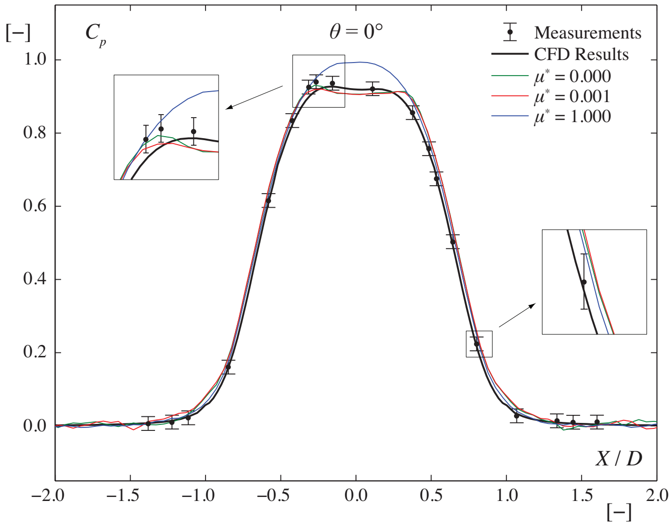


Figure 4.7 – Influence of the artificial viscosity on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$. The artificial viscosity influence is negligible when $\mu^* \le 0.001$ Pa·s.

4.3 Results

4.3.1 Pressure profile

The influence of spatial and time discretizations are investigated in the case of a $\theta = 0°$ impinging angle. The computing hours and final number of particles are summarized in Table 4.1. According to this table, the elapsed time of a simulation is strongly influenced by the number of particles, and so, by the prescribed spatial discretization. The time discretization affects only the computing time of the simulation. However, instabilities appear during the computation when this value is larger than 0.7. Consequently, a maximum CFL of 0.5 is selected for the following analyses. In the cases where the spatial discretization is not specifically mentioned, the medium discretization $D/X_{\text{ref}} = 20$ is selected to ensure that the computational time remains below 5 hours.

Table 4.1 – Influence of spatial and time discretizations.

D/X_{ref}	max(CFL)	Computing hours	N_{final}
10	0.5	0.89	41’204
20	0.5	3.93	197’255
40	0.5	81.92	1’024’971
20	0.3	6.14	198’025
20	0.5	3.93	197’255
20	0.7	3.43	196’771

The influence of the spatial discretization is analyzed for three different impinging angles; $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. The averaged C_p along the X axis of the flat plate are given in Figure 4.8. The global shape of the pressure profiles is extremely well represented for all the impinging angles. However, the finer discretization $D/X_{\text{ref}} = 40$ is required to capture properly the very sharp pressure profile for the $\theta = 60°$ impinging angle. By increasing the spatial discretization, the number of particles in the high gradient zone is increased, which improves the FPM results and increases the computing time.

4.3.2 Free surface location

The free surface location is compared in Figure 4.9 for the three different impinging angles; $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. Despite some small oscillations with the coarser resolution, all the FPM simulations present a good accuracy to capture the free surface location compared to the measurements and CFD results. For the $\theta = 60°$ impinging angle, the free surface location is only compared to the CFD results because the experimental arrangement did not allow to measure accurately the free surface location. A 3-D visualization of the FPM results is presented in Figure 4.10. The particles data are first projected on a unstructured grid. Then, the free surface is computed and exported as a unstructured surface. The rendering is done with the software *POV-Ray*, which defines the camera location, light effects and external components in order to generate the images.

4.3.3 FPM vs FVPM simulations

The averaged C_p along the X axis of the flat plate is presented in Figure 4.11 for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$ and for both FPM and FVPM results. In these simulations, the spatial discretization corresponds to $D/X_{\text{ref}} = 40$ for both methods. Although the FVPM is not able to capture properly the pressure peak at 60°, the FVPM fits better the CFD results and measurements compared to FPM. However, the FVPM computing time is twice bigger compared to the FPM computing time. This drawback can be mitigated by increasing the domain decomposition, i.e. increasing the number of cores required for the simulation.

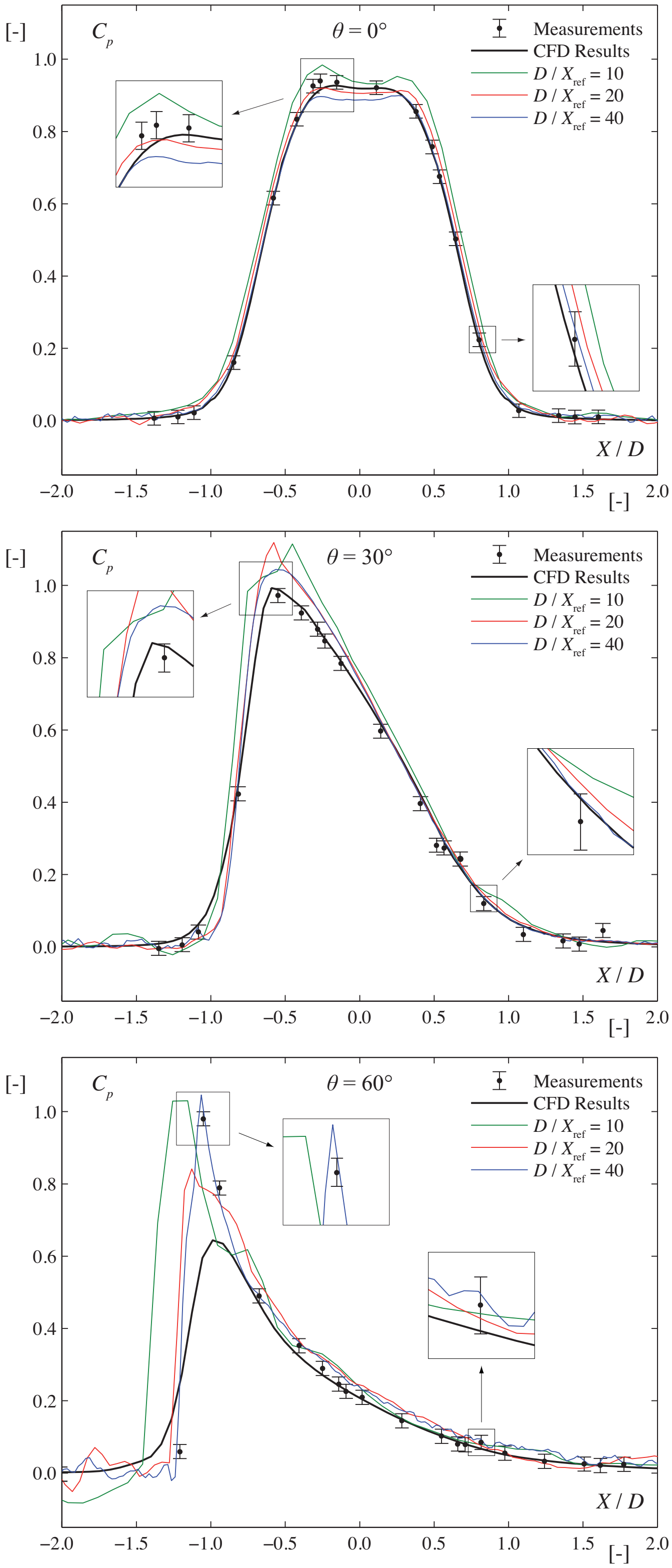


Figure 4.8 – Influence of the spatial discretization on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$.

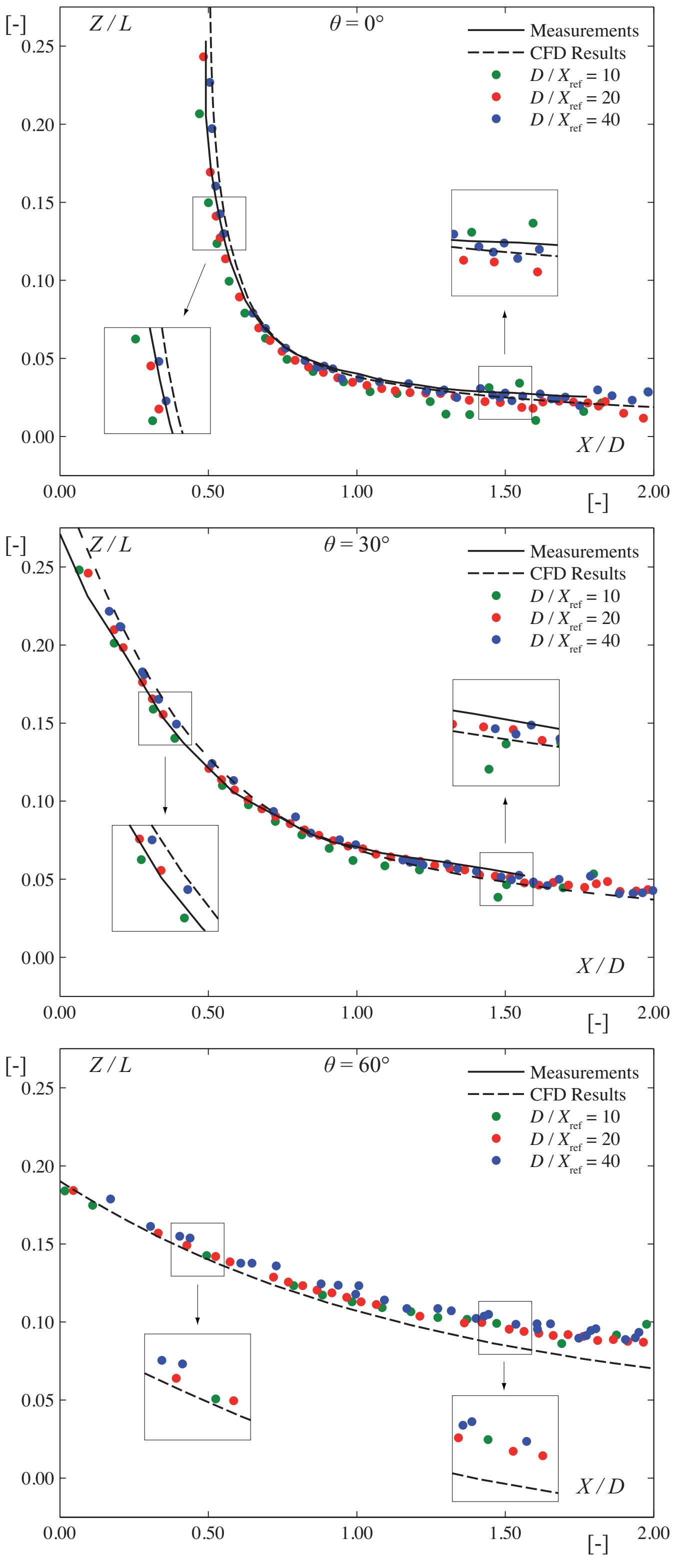
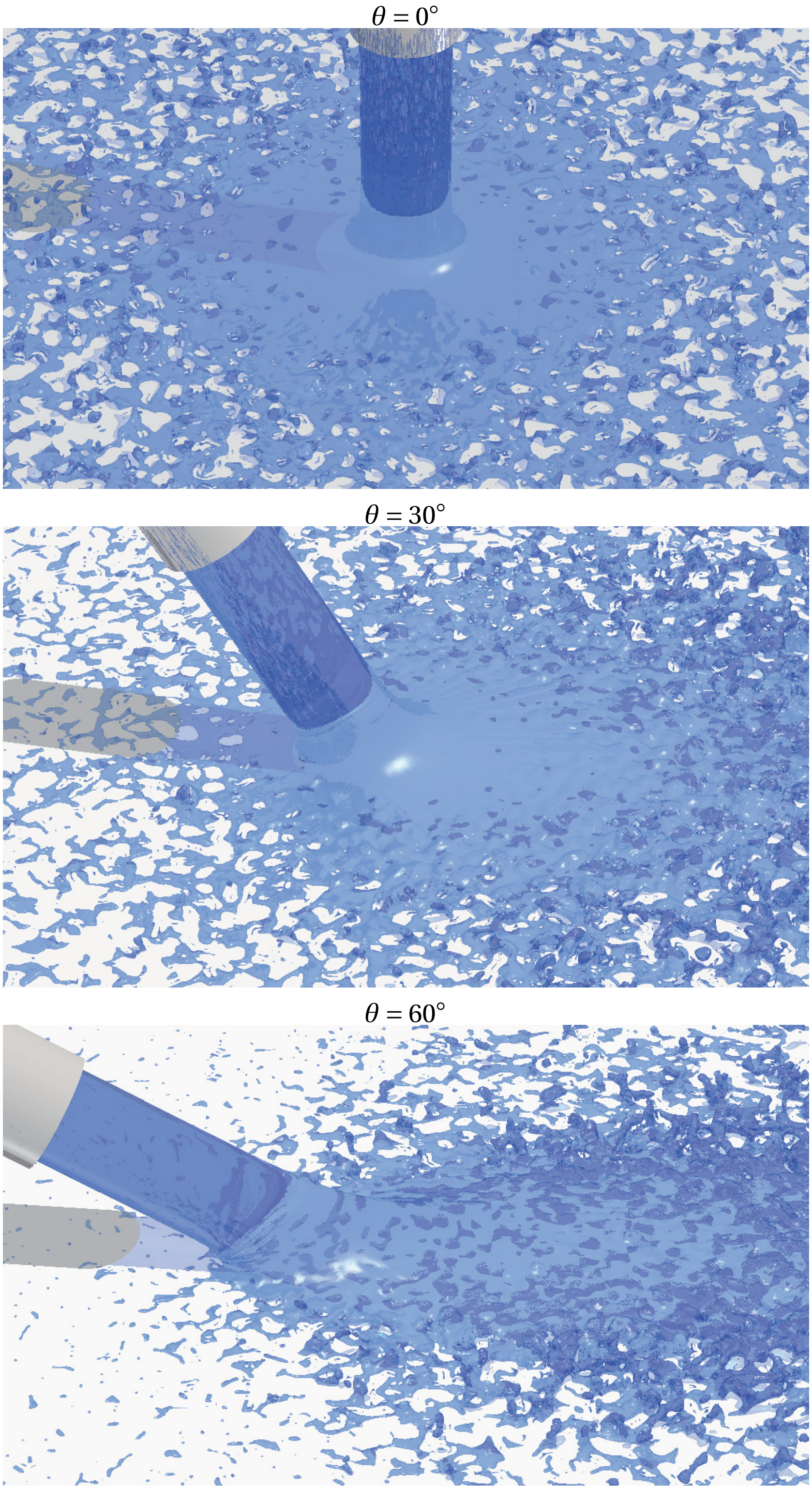


Figure 4.9 – Influence of the spatial discretization on the free surface location in the XZ plane for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$.


$\theta = 0^\circ$

$\theta = 30^\circ$

$\theta = 60^\circ$

Figure 4.10 – Rendering of the FPM simulations of a $\theta = 0^\circ$, $\theta = 30^\circ$ and $\theta = 60^\circ$ impinging jet using the software *POV-Ray*.

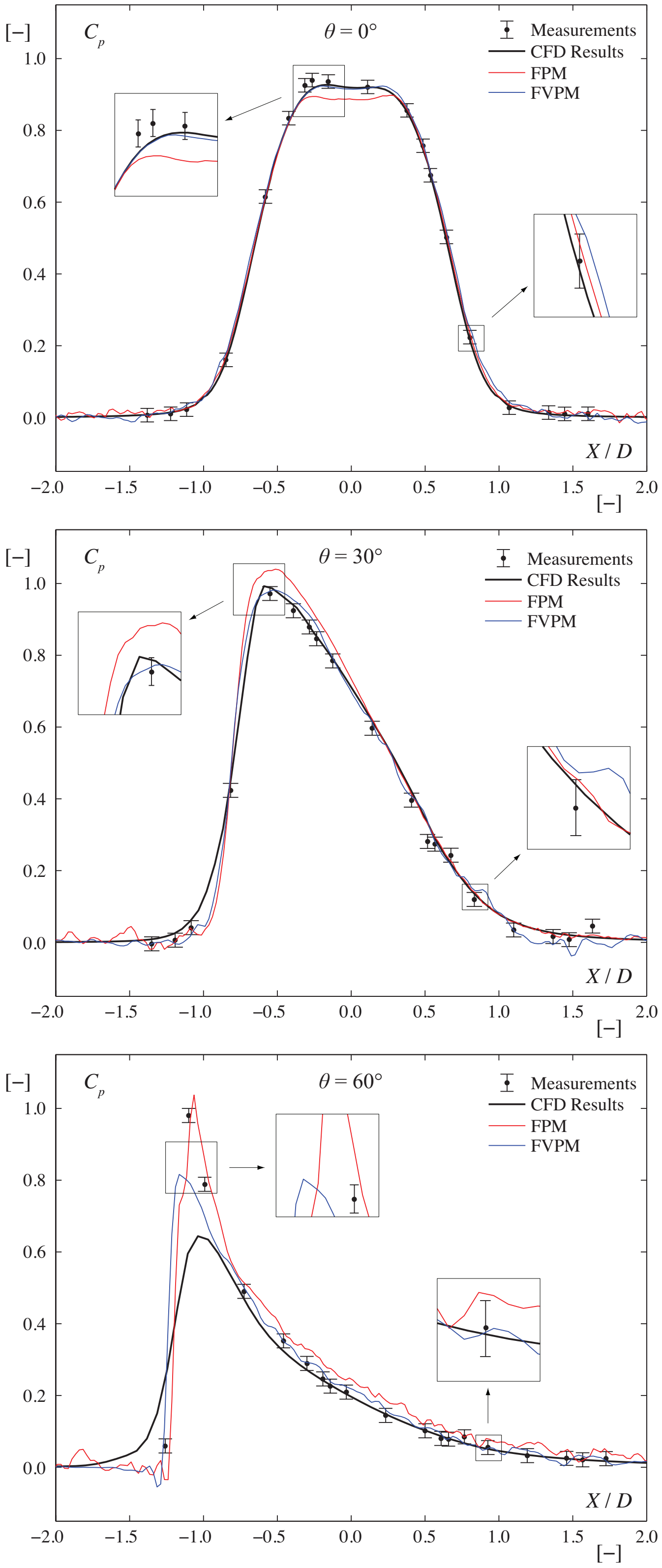


Figure 4.11 – Comparison between FPM and FVPM on the averaged C_p along the X axis of the flat plate for an impinging angle of $\theta = 0°$, $\theta = 30°$ and $\theta = 60°$. The spatial discretization corresponds to $D/X_{\text{ref}} = 40$ for both methods.

5 Impinging Jet on Pelton Buckets

5.1 Steady bucket analysis

5.1.1 Case study

The purpose of the present section is to validate that FVPM simulations are able to capture accurately the deviation of a high-speed water jet by a stationary Pelton bucket. The FVPM simulations are done with the software *SPHEROS*, developed at EPFL since 2010 [32, 75]. The FVPM simulations of the flow in a stationary bucket are validated by VOF numerical simulations and experimental data obtained by Kvicinsky et al. [39]. The wall pressure field in the stationary bucket inner wall is used to compare the FVPM results with pressure sensors and VOF results.

In the present study, a high-speed water jet impinges on a stationary bucket at different impinging angles. An outline of the case study is given in Figure 5.1. The bucket geometry and the 32 pressure samples location are taken from Kvicinsky et al. [39]. The bucket width is $B_2 = 0.09$ m and its reference diameter is $D_1 = 0.315$ m. The location of the pressure samples fits the position of lines $X1$ to $X7$, $Z1$ to $Z5$ and $D1$ to $D4$. In this stationary analysis, the bucket is tilted of an angle $\theta = 72°$, $90°$ or $108°$ around the Z axis. The jet diameter is $D_2 = 0.03$ m and its axis is in the $-X$ direction at a distance $Y = -D_1/2$. The discharge velocity of the water jet is $C_{\max} = 28.5\ \mathrm{m\,s^{-1}}$. The discretization of the bucket geometry into wall particles is performed using the software *ANSYS ICEM*. An unstructured triangular mesh is applied to the bucket inner and outer walls. The maximum edges size of the 2-D triangular mesh are limited to X_{ref}. The node location are imported into *SPHEROS* to model the bucket geometry and the connectivity table is stored for the post-processing of the results.

The FVPM simulations are run on two Intel Xeon CPUs E5 2670 at 2.6 GHz with 32 cores (hyper-threading 2×16) and 32 Gb of memory. The domain decomposition is $4 \times 4 \times 4$ and the domains size is adapted according to the particles load using the adaptive domain decomposition strategy from Vessaz et al. [76].

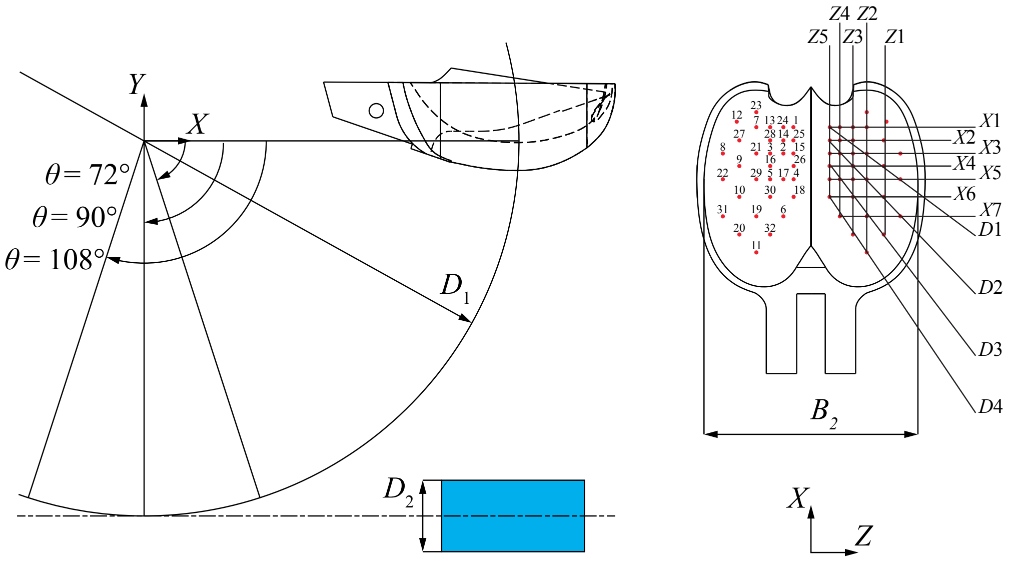


Figure 5.1 – Outline of the case study. The bucket can tilt of an angle θ around the Z axis. The inlet of the water jet has a diameter D_2 and its axis is in the $-X$ direction at a distance $Y = -D_1/2$. The 32 pressure samples are located on the bucket inner surface, their location fits the position of lines $X1$ to $X7$, $Z1$ to $Z5$ and $D1$ to $D4$.

5.1.2 Convergence study

The influence of the reference particle spacing X_{ref} on the FVPM results is analyzed for an impinging angle of $\theta = 90°$. The time history of F/F^* is shown in Figure 5.2, where F is the magnitude of the force applied on the bucket and $F^* = 2\rho\pi(D_2/2)^2 C_{\text{max}}^2$ is the maximum force of the water jet. A mean value of the converged force, the computing time as well as the number of particles at the end of the simulation are given in Table 5.1.

Table 5.1 – Influence of the spatial discretization.

D_2/X_{ref}	F/F^*	Computing hours	N_{final}
10	0.8226	6.17	9’075
20	0.8761	5.55	51’174
30	0.8956	9.55	146’495
40	0.9097	20.92	317’688
50	0.9188	44.33	584’336

According to these results, the convergence of the FVPM results with the refinement of the spatial discretization is highlighted. However, the computing cost increases significantly with the spatial discretization. A free surface reconstruction of the water sheet is shown in Figure 5.3 for a discretization of $D_2/X_{\text{ref}} = 30$. The free surface location is less influenced by the spatial discretization compared to the force or pressure measurements.

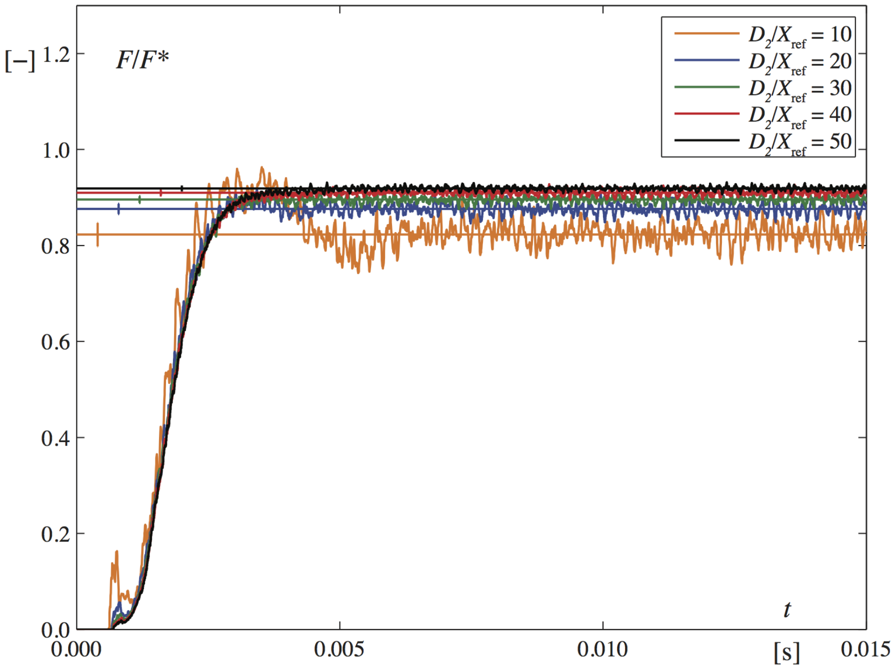


Figure 5.2 – Time history of the force in a $\theta = 90°$ stationary bucket: influence of the spatial discretization.

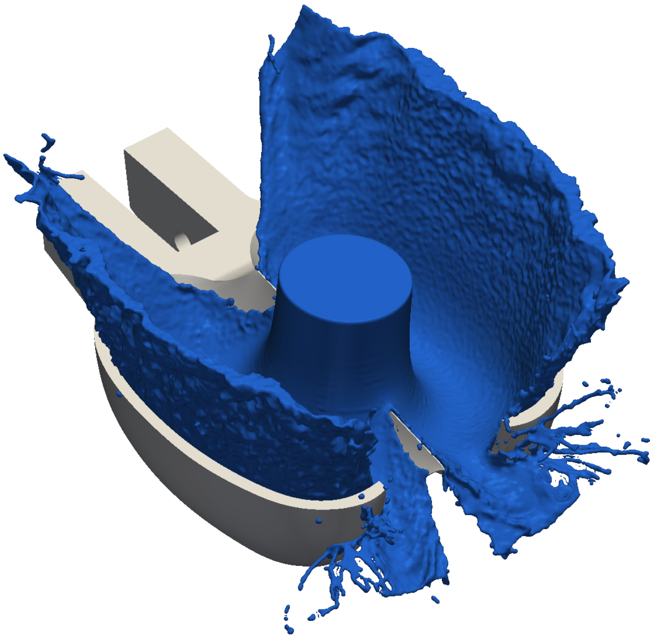


Figure 5.3 – FVPM simulation in a stationary bucket at an impinging angle of $\theta = 90°$: free surface reconstruction using *ParaView*.

5.1.3 Results

The pressure coefficient $C_p = (p - p_{\text{ref}})/(0.5\rho C_{\text{max}}^2)$ is compared to the VOF computations and measurements from Kvicinsky et al. [39]. The averaged C_p profile along the lines $X1$ to $X7$, $Z1$ to $Z5$ and $D1$ to $D4$ are given in Figures 5.4, 5.5 and 5.6 respectively. Three different discretizations are presented for the FVPM results: coarse $D_2/X_{\text{ref}} = 10$, medium $D_2/X_{\text{ref}} = 30$ and fine $D_2/X_{\text{ref}} = 50$.

The FVPM pressure profiles on the lines $X1$ to $X7$ fit qualitatively well the VOF and measurements from Kvicinsky et al. [39] despite some quantitative differences. Moreover, the VOF results present also some quantitative differences in comparison to the measurements, e.g. lines $X1$ and $X2$. The difference between the medium and fine FVPM simulations is very small, which highlights the convergence of the FVPM results according to the spatial discretization. The FVPM results fits better the measurement close to the impingement zone, i.e. lines $X3$ to $X5$ and for the values of $Z \leq D_2$.

The convergence of the pressure profile is also highlighted by comparing the medium and fine FVPM results on the lines $Z1$ to $Z5$. Moreover, the FVPM results fit well the measurements from Kvicinsky et al. [39]. As highlighted in the comparison on the lines $X1$ to $X7$, the FVPM pressure profiles are better predicted close to the impingement zone, i.e lines $Z3$ to $Z5$ and for the values of $4.75D_2 \leq X \leq 5.75D_2$.

The pressure profile for the diagonal lines $D1$ to $D4$ are represented according to L, which represents the distance between the tip of the bucket and the sample location in the plane XZ. The convergence of the FVPM results is also highlighted in these diagonal lines and the shape of the pressure profiles fit well the measurements.

The wall pressure field comparisons between the finest FVPM results of $D_2/X_{\text{ref}} = 50$, VOF and measurements are presented in Figure 5.9 for the three different impinging angles $\theta = 72°$, $\theta = 90°$ and $\theta = 108°$. The measurements wall pressure field is rough because it is interpolated by only 32 pressure samples location. The FVPM wall pressure field features more numerical fluctuations compared to VOF. However, these fluctuations decrease when the spatial resolution is increased (see Figure 5.2). The highest value of C_p is found on the splitter due to the bucket discretization for both VOF and FVPM. Generally speaking, the FVPM results for the three different impinging angles fit qualitatively well the VOF computations and measurements. However, some differences are highlighted at $\theta = 72°$ close to the cut out location. Finally, FVPM results fit better the measurement than VOF in the center of the bucket inner wall. Indeed, in this zone the VOF pressure field is more diffusive than the FVPM results.

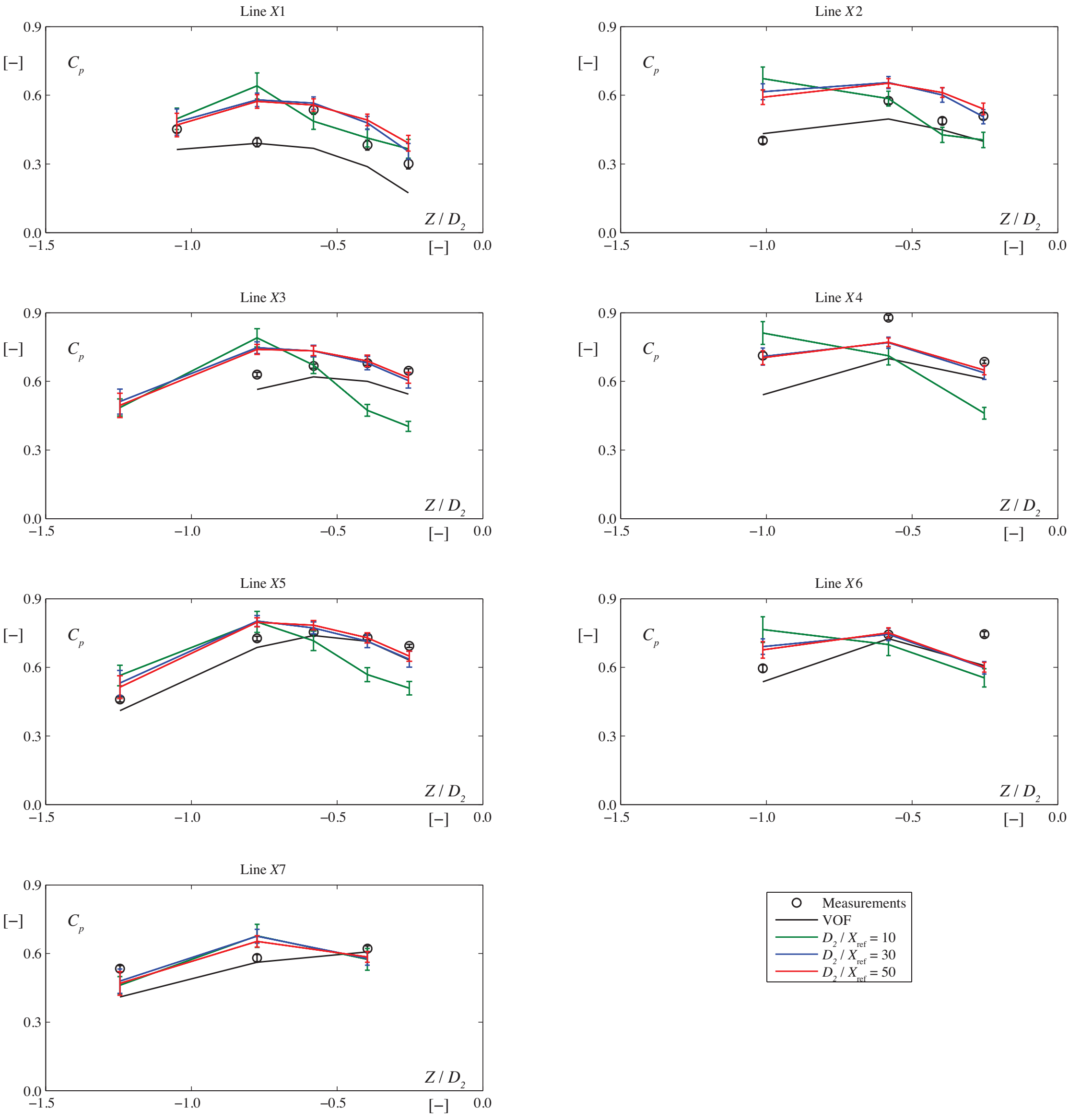


Figure 5.4 – Averaged C_p along the lines $X1$ to $X7$ of the bucket at an impinging angle of $\theta = 90°$: influence of the spatial discretization.

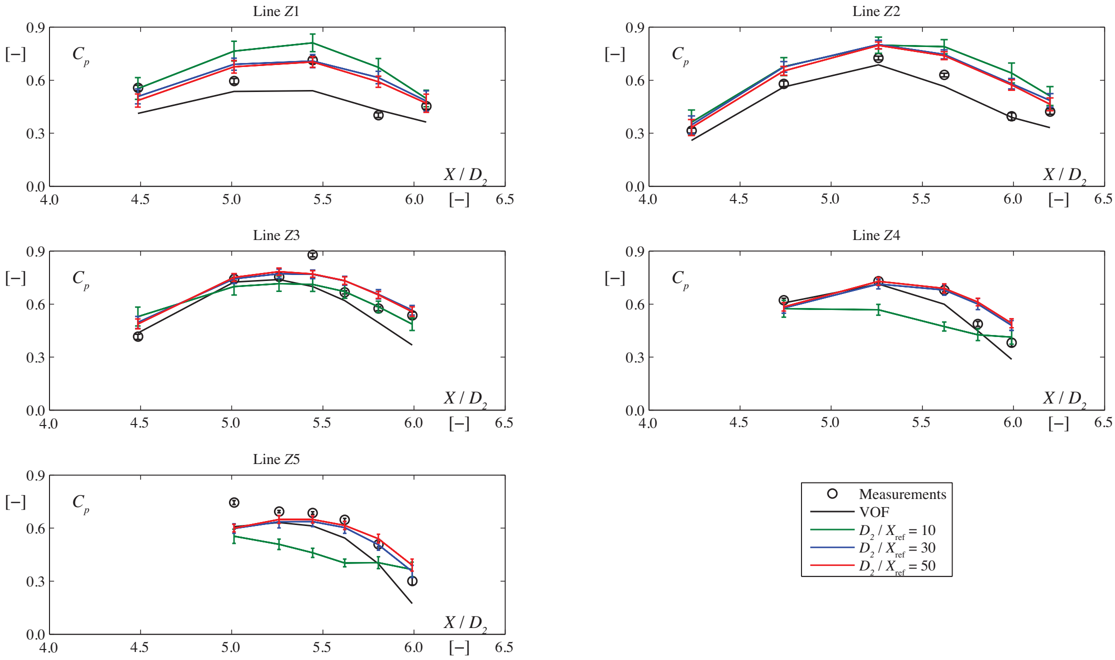


Figure 5.5 – Averaged C_p along the lines $Z1$ to $Z5$ of the bucket at an impinging angle of $\theta = 90°$: influence of the spatial discretization.

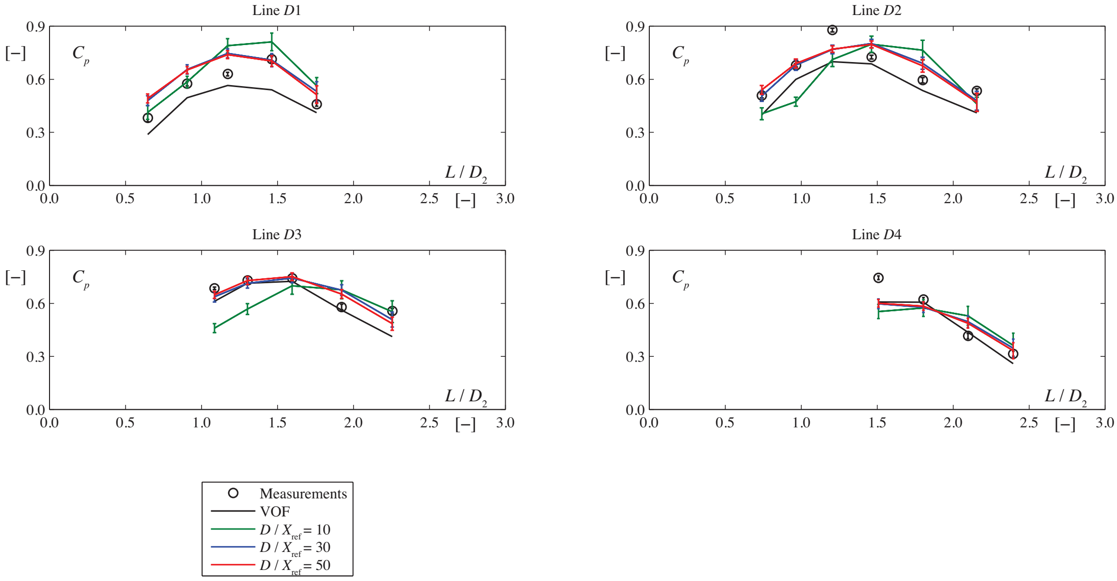


Figure 5.6 – Averaged C_p along the lines $D1$ to $D4$ of the bucket at an impinging angle of $\theta = 90°$: influence of the spatial discretization.

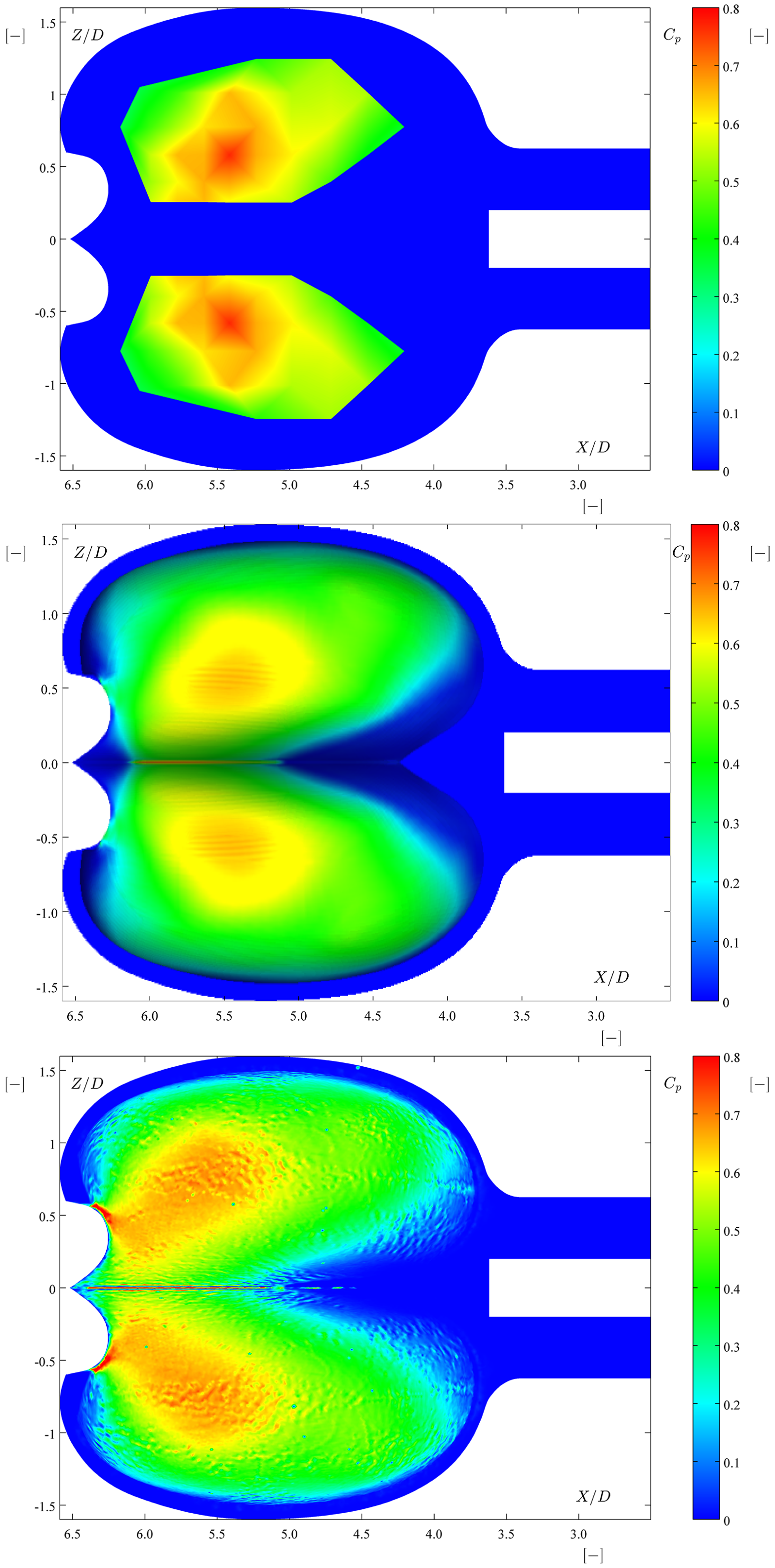


Figure 5.7 – Comparison of the wall pressure field between experimental (up), VOF (middle) and FVPM (down) for the impinging angles $\theta = 72°$.

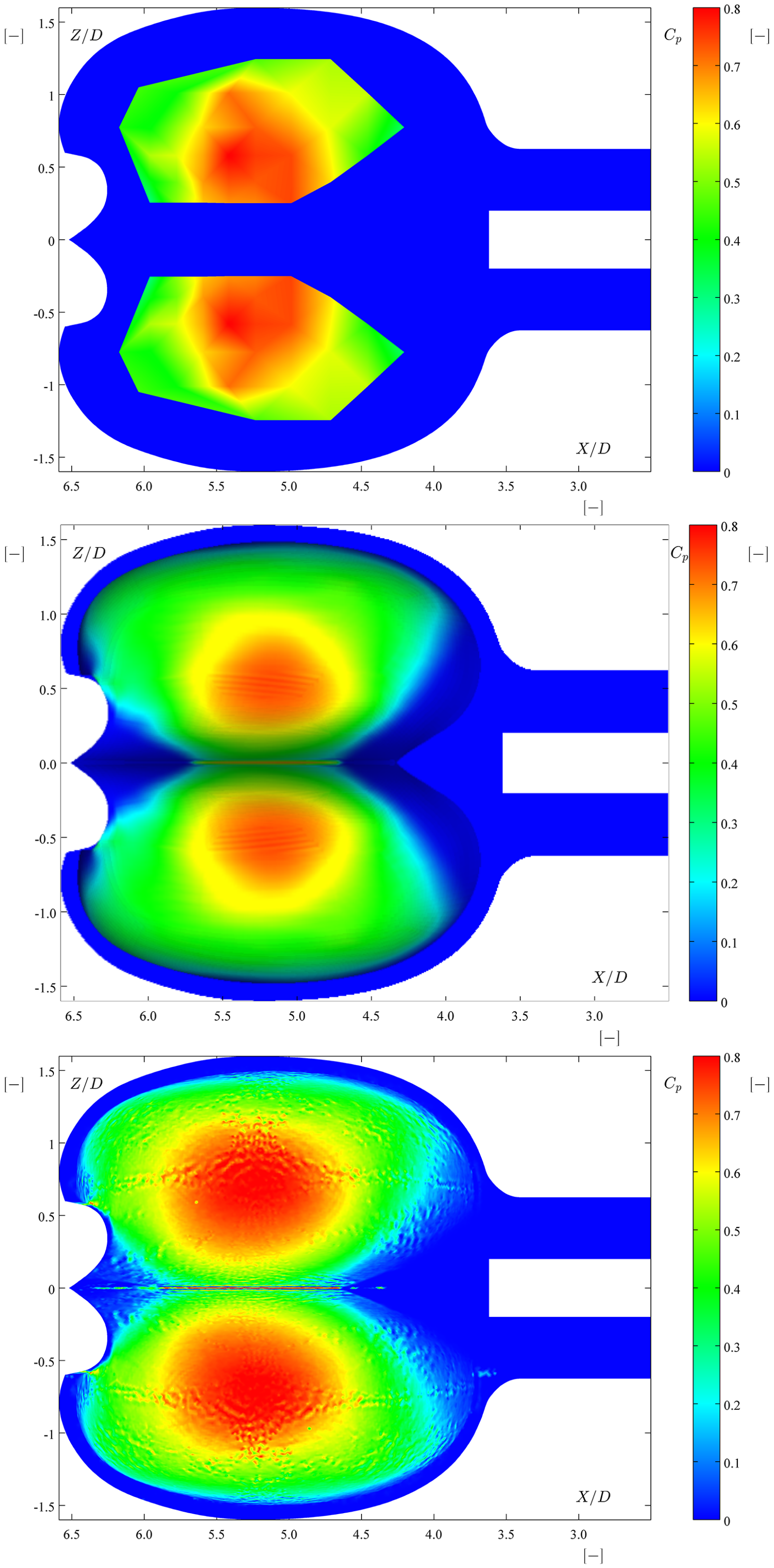


Figure 5.8 – Comparison of the wall pressure field between experimental (up), VOF (middle) and FVPM (down) for the impinging angles $\theta = 90°$.

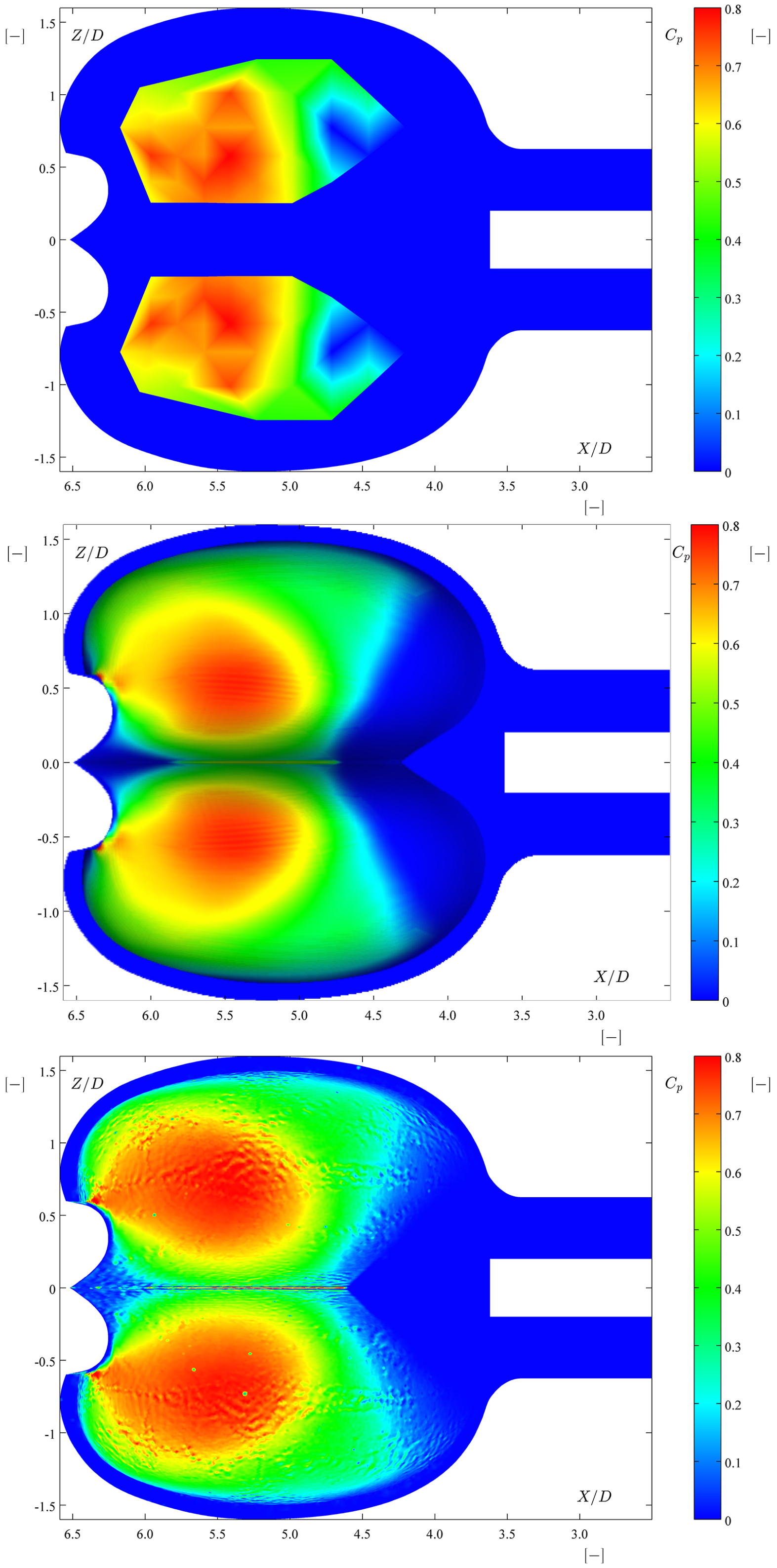


Figure 5.9 – Comparison of the wall pressure field between experimental (up), VOF (middle) and FVPM (down) for the impinging angles $\theta = 108°$.

5.2 Rotating buckets analysis

5.2.1 Case study

The purpose of the present section is to validate that FVPM simulations are able to capture accurately the deviation of a high-speed water jet by five rotating Pelton buckets. The FVPM simulations are done with the software *SPHEROS*, developed at EPFL since 2010 [32, 75]. The FVPM simulations are validated with VOF numerical simulation and experimental data obtained by Perrig [65].

In the present study, a high-speed water jet impinges on five rotating buckets. An outline of the case study is given in Figure 5.10 and 5.11. The buckets geometry and the 43 pressure samples location are taken from Perrig [65]. The bucket width is $B_2 = 0.08$ m and its reference diameter is $D_1 = 0.328$ m. Five buckets rotate around the Z axis with a rotation speed $N = 1'025$ rpm. The five buckets are spaced with an angle of 17.14° corresponding to a Pelton runner featuring 21 buckets. The water jet is oriented in the $-X$ direction and its inlet is located at $X = 0.185$ m, $Y = -D_1/2$ and $Z = 0$ m. The operating point simulated corresponds to a discharge coefficient $\varphi_{B_2} = 0.233$ and an energy coefficient $\psi_1 = 3.8$. The discharge velocity of the water jet is $C_{\max} = 33.6\ \mathrm{m\,s^{-1}}$ with a diameter $D_2 = 0.028$ m. At the initial condition, the first bucket has an angle of $\theta = 54°$ and the length of the water jet is 0.1 m. The results are saved each 0.5042° and the simulation time corresponds to a 160° buckets rotation angle.

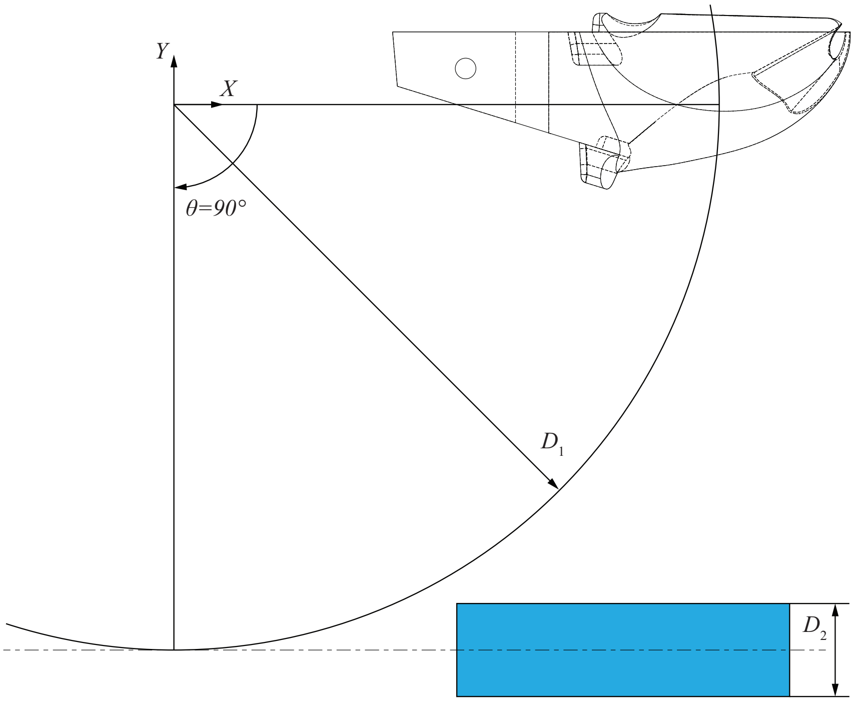


Figure 5.10 – Outline of the case study. The buckets are defined by a reference diameter D_1 and a width B_2. They tilt around the Z axis and their angular position is given by θ. The inlet of the water jet has a diameter D_2 and its axis is in the $-X$ direction at a distance $Y = -D_1/2$.

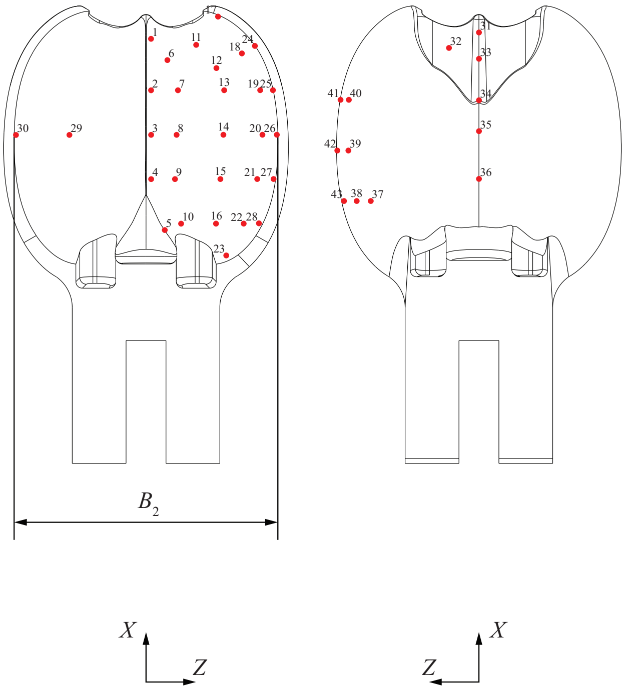


Figure 5.11 – Outline of the case study. 30 pressure samples are located on the buckets inner wall and 13 on the outer wall.

The discretization of the buckets geometry into wall particles is performed using the software *ANSYS ICEM*. An unstructured triangular mesh is applied to the buckets inner and outer walls. The maximum edges size of the 2-D triangular mesh are limited to X_{ref}. The node location are imported into *SPHEROS* to model the buckets geometry and the connectivity table is stored for the post-processing of the results. In the case of an arbitrary Pelton bucket design, the appendix A proposes a parametric model to generate the buckets geometry using 4 Bézier surfaces for the inner wall and an offset method for the outer wall.

The FVPM simulations are run on the *Lemanicus* IBM BG/Q at EPFL, which features one rack of 1'024 sixteen-cores computing nodes PowerA2 at 1.6 GHz, 16 GB of memory per computing node and a 5-D torus communication network. The domain decomposition depends on the spatial discretization of the simulation. The domains size is adapted according to the particles load using the adaptive domain decomposition strategy developed by Vessaz et al. [76].

5.2.2 Convergence study

Spatial resolution

The influence of the reference particle spacing X_{ref} on the resulting torque is investigated in Figure 5.12. The evolution of the torque coefficient $T_{ED} = T/(\rho \cdot E \cdot D_1^3)$ in each bucket as well as the total torque applying on the five buckets in function of the angular position is shown in this figure for the finest particles resolution $D_2/X_{\text{ref}} = 50$. The convergence of the FVPM results is highlighted in Figure 5.13, where the evolution of the averaged torque in one bucket is presented. This averaged torque is obtained by time averaging the torque in the buckets 2, 3 and 4. The fluctuation of the torque signal decreases when the particles resolution is increased. Moreover, the onset of the first pressure peak converges to an angle of 72° by increasing the spatial discretization. Therefore, the FVPM simulations with a particles resolution of $D_2/X_{\text{ref}} = 40$ and $D_2/X_{\text{ref}} = 50$ are considered as converged.

The domain decomposition, the mean computing time per step, the number of time step as well as the number of particles at the end of the simulation are given in Table 5.2 for each particles resolution. The domain decomposition is adapted according to the spatial discretization in order to obtain an initial number of particle per core around 500. The adaptive domain decomposition strategy adapts the domains size during the computation to ensure a maximum load difference of 20% between the subdomains. Therefore, the influence of the particles number on the computing time per time step is decreased. However, the number of time steps is directly linked to the particles spacing due to the CFL definition given in (2.74). A free surface reconstruction of the water sheet for a discretization of $D_2/X_{\text{ref}} = 40$ is presented in Figure 5.14. The free surface location is less influenced by the spatial discretization compared to the torque or pressure measurements.

Table 5.2 – Influence of the spatial discretization.

D_2/X_{ref}	**Decomposition**	**Time per step**	N_{step}	N_{final}
10	4 × 4 × 4	30.1	7’262	60’167
20	16 × 4 × 4	10.2	14’580	211’307
30	24 × 5 × 5	16.2	21’814	582’261
40	32 × 6 × 6	18.1	29’293	1’243’887
50	38 × 7 × 7	20.5	36’886	2’278’456

Sound speed effect

The influence of the sound speed on the resulting torque is investigated in Figure 5.15 for a spatial discretization which corresponds to $D_2/X_{\text{ref}} = 40$. The black curve represents the FVPM results with the standard value of sound speed $a = 10 \cdot C_{\text{max}}$. The green curve highlights that the pressure peak at $\theta = 72°$ is not captured when the sound speed is lower. However, increasing the sound speed does not improve the results (see the red curve). Indeed, the simulation parameters such as CFL or artificial viscosity have been tuned for the sound speed of $a = 10 \cdot C_{\text{max}}$. Moreover, increasing the sound speed will decrease the time step value, which will increase the computing time.

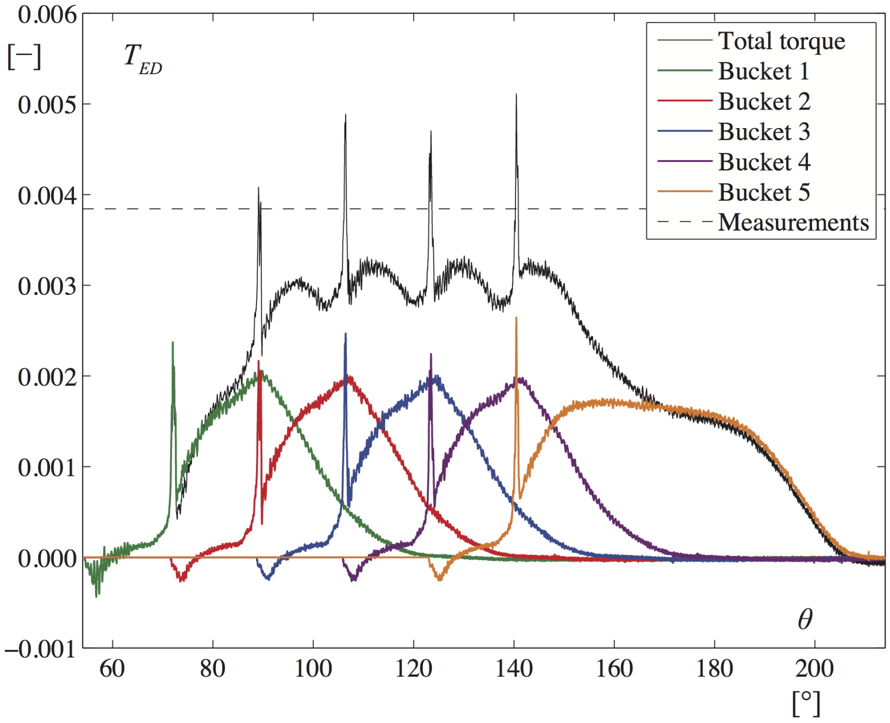


Figure 5.12 – Evolution of the torque for each bucket (colors) and total torque (black) in function of the angular position.

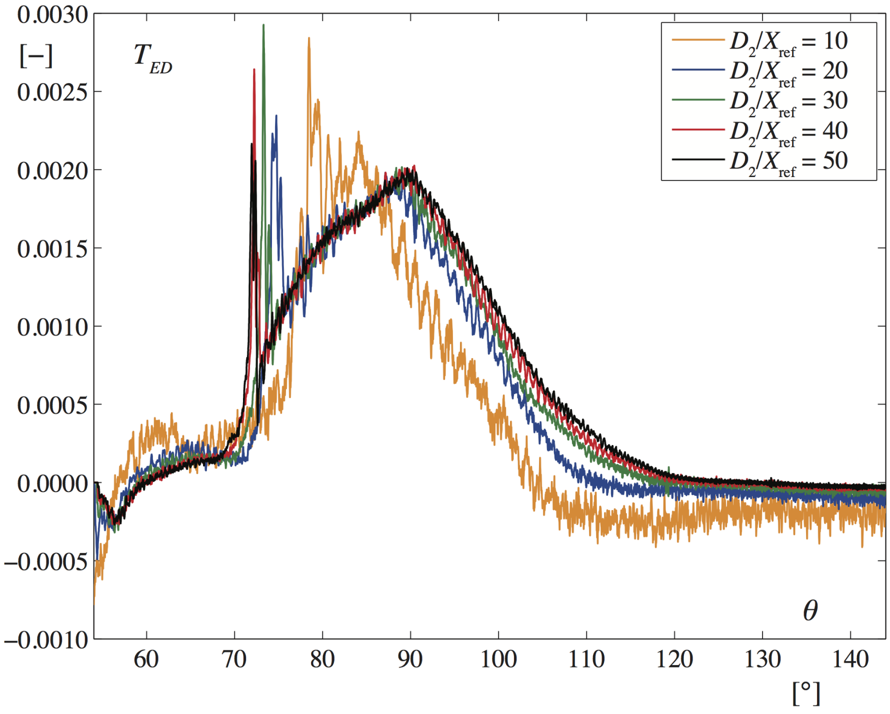


Figure 5.13 – Averaged torque for one bucket in function of the angular position: influence of the spatial resolution.

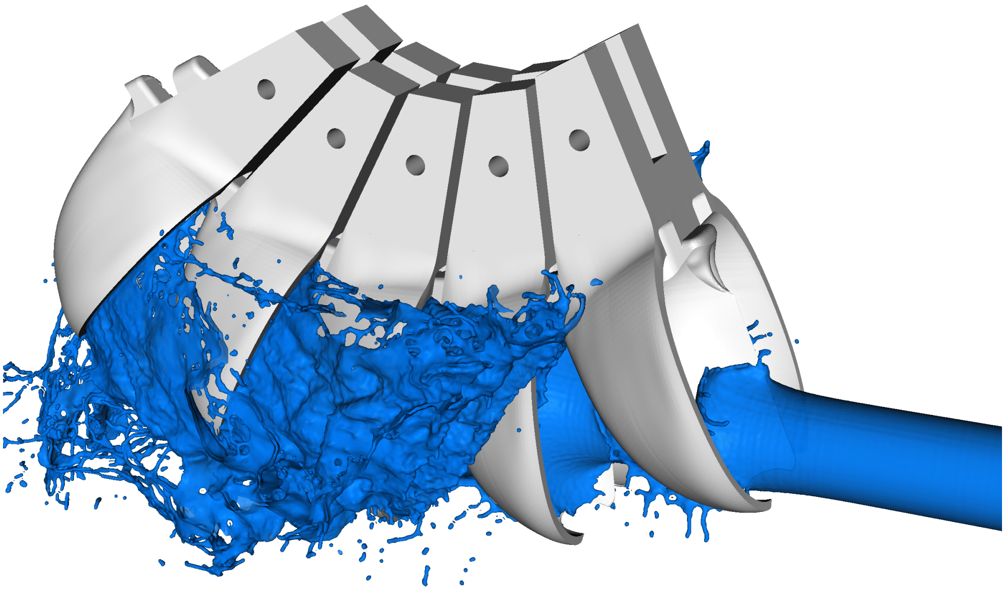


Figure 5.14 – FVPM simulation of five rotating buckets: free surface reconstruction using *ParaView*.

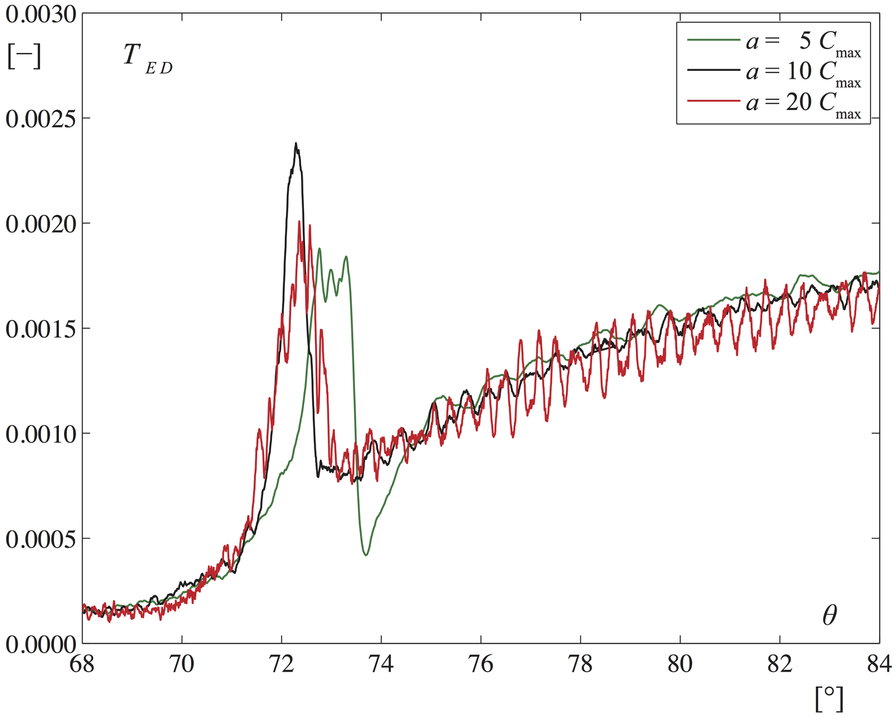


Figure 5.15 – Averaged torque for one bucket in function of the angular position: influence of the sound speed.

5.2.3 Results

Pressure samples

The pressure coefficient $C_p = (p - p_{\text{ref}}) / (0.5 \rho C_{\text{max}}^2)$ is compared to the VOF computation and measurements from Perrig [65]. The FVPM results are obtained by averaging C_p in the buckets 2, 3 and 4 for the discretization $D_2 / X_{\text{ref}} = 50$. The time history of C_p for the 43 pressure sensors are given in Figures 5.16 to 5.22, which correspond to the different zones in the bucket defined by Perrig et al. [66]. On the inner surface, five different zones are highlighted: the splitter side of the frontal region, the central region, splitter side of the aft region, the outer side of the aft region and the outer side of the frontal region. On the outer surface, the pressure samples are divided in two zones: the splitter edge region and the outer edge region. The RMS fluctuation of the pressure signals are represented by the light colored area in Figures 5.16 to 5.22.

On the splitter side of the frontal region (see Figure 5.16), the shape of the FVPM pressure signals fits qualitatively well the VOF computation and measurements except for the sample 1. However, the sample 1 is located close to the bucket tip and does not contribute significantly to the torque generation. The numerical fluctuations of the FVPM signals are always larger compared to the RMS fluctuation of the VOF signals but the impingement early stage is better predicted with the FVPM results compared to the VOF results, see samples 6 and 7. The tail of pressure profile is underestimated between $\theta = 90^\circ$ and $\theta = 120^\circ$ for the FVPM signals in sample 3 and overestimated in samples 6 and 11.

The pressure peak at $\theta = 72^\circ$ is highlighted in the central region (see Figure 5.17). This peak is also highlighted in the measurement for the sample 14 but the VOF results only capture a small disturbance of the pressure signals. Despite a small underestimation of the tail of pressure profile between $\theta = 90^\circ$ and $\theta = 130^\circ$, the FVPM results fit qualitatively well the VOF computation and measurements in this central region.

On the splitter side of the aft region (see Figure 5.18), the samples 4, 5 and 10 feature only numerical fluctuations due to some water droplets stuck close to the bucket root. The shape of the pressure samples 8, 9 and 16 fits qualitatively well the VOF computation and measurements but the tail of pressure profile is also underestimated between $\theta = 90^\circ$ and $\theta = 120^\circ$. The pressure peak at $\theta = 72^\circ$ is only seen by samples 8 and 9 and its magnitude is smaller compared to the samples located in the central region.

On the outer side of the aft region (see Figure 5.19), the shape of the pressure signals fits qualitatively well the VOF computation and measurements but the RMS fluctuations of FVPM results are always larger than the RMS fluctuations of VOF results. The VOF and FVPM pressure signals for pressure sample 23 have difficulties to match the measurements, which can be explained by the lack of water close to the bucket root.

On the outer side of the frontal region (see Figure 5.20), the shape of the pressure signals fits qualitatively well the VOF and measurements. However, the impingement early stage is predicted 5° in advance for the samples 17, 18 and 24. The numerical pressure fluctuations are also larger for the FVPM results compared to the VOF results.

On the splitter edge of the outer surface (see Figure 5.21), the VOF computation fits better the measurements than the FVPM results. The pressure sample 31 is the only one which can reproduce the shape of the measurements pressure profile. However, the FVPM results highlight significant numerical fluctuations compared to the bucket inner surface. The VOF results feature also some unexpected fluctuations between $\theta = 100°$ and $\theta = 140°$.

On the outer edge of the outer surface (see Figure 5.22), both VOF and FVPM have difficulties to capture the measurements. This difficulty to represent properly the pressure profile on the outer edge of the outer surface is explained by the lack of particles close to the sampling locations. The pressure sample 37 is not represented in Figure 5.22 because all the pressure signals feature a constant zero value.

Consequently, the shape of the FVPM pressure signals on the buckets inner wall fits qualitatively well the VOF computation and measurement from Perrig [65]. Moreover, the FVPM simulation is able to capture the pressure peak at $\theta = 72°$, e.g. pressure sample 14. However, the numerical fluctuations of the FVPM signals are always larger than the RMS fluctuations of the VOF results. These numerical fluctuations are partly explained by the lack of particles close to the sampling location, e.g. the buckets outer wall.

Wall pressure field

The FVPM wall pressure field on the buckets inner wall is presented in Figure 5.23 to Figure 5.26 for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°, \theta = 72°, \theta = 77°, \theta = 82°, \theta = 87°, \theta = 94°, \theta = 99°, \theta = 104°, \theta = 109°, \theta = 104°, \theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$. These impinging angles have been chosen for the comparison of the free surface evolution in the following section.

At $\theta = 57°$, the bucket features the impingement early stage: the pressure field increases on the bucket tip and cutout. At $\theta = 62°$ and $\theta = 67°$, the wall pressure field moves along the splitter. The pressure peak at $\theta = 72°$ is highlighted in Figure 5.23. According to Perrig [65], this pressure peak reached during this initial impact stage depends on the contact area edge expansion Mach number, the liquid density and the local sound speed in the liquid. Therefore, the weakly compressible approach of FVPM is better suited to capture this phenomenon than the incompressible approach of VOF. Between $\theta = 77°$ and $\theta = 87°$, the central and aft region of the bucket feature the highest wall pressure field. Moreover, the FVPM wall pressure field highlights the impingement of many water droplets (see Figure 5.24), which explains the fluctuations present in the sampling comparisons in Figures 5.16 to 5.22. Between $\theta = 94°$ and $\theta = 114°$, the frontal region features the highest wall pressure field (see Figure 5.25). The remaining water droplets are evacuated between $\theta = 119°$ and $\theta = 134°$ (see Figure 5.26).

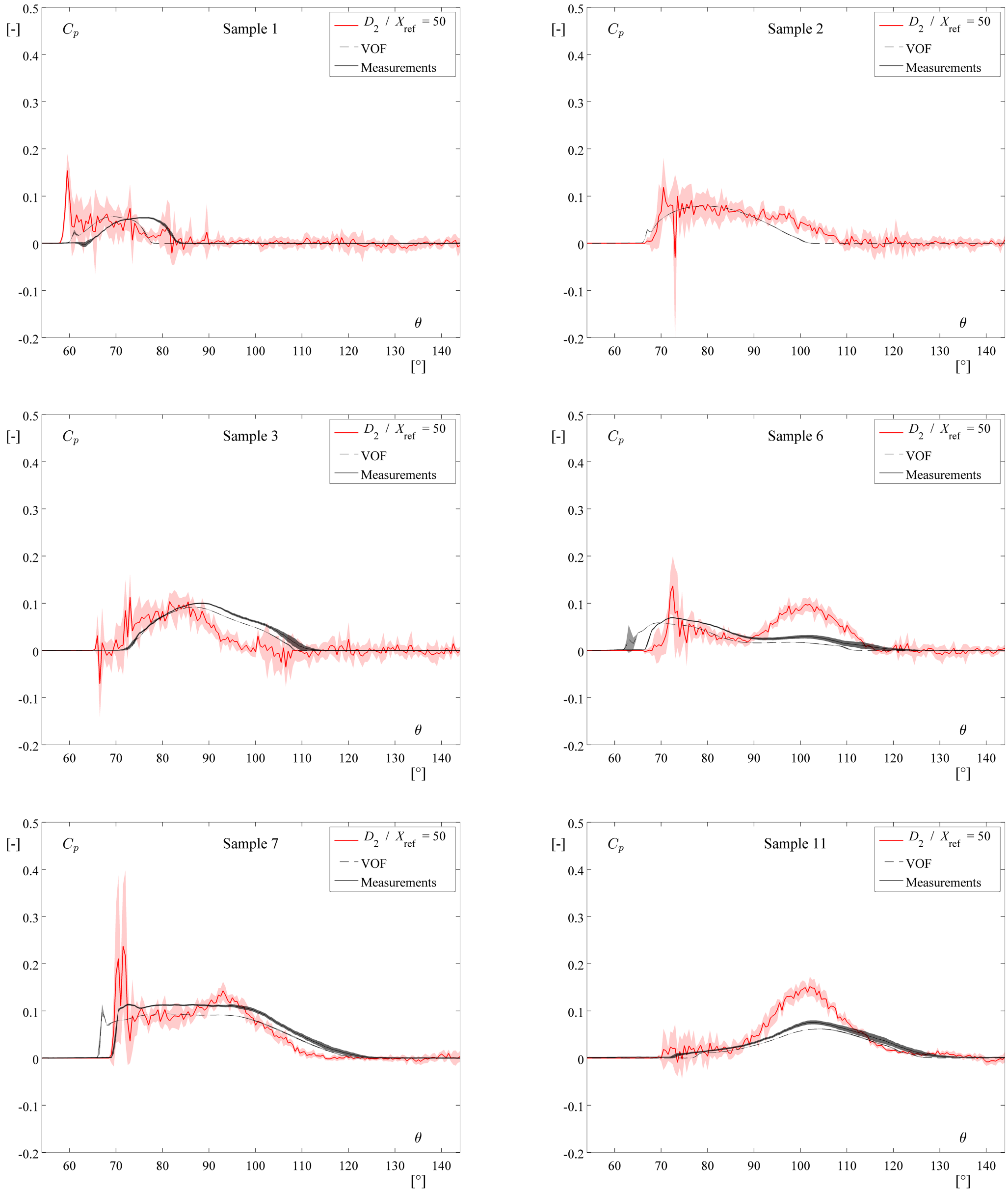


Figure 5.16 – Time history of C_p at the pressure sensors 1, 2, 3, 6 and 7 on the buckets inner surface, which corresponds to the splitter side of the frontal region of the inner surface.

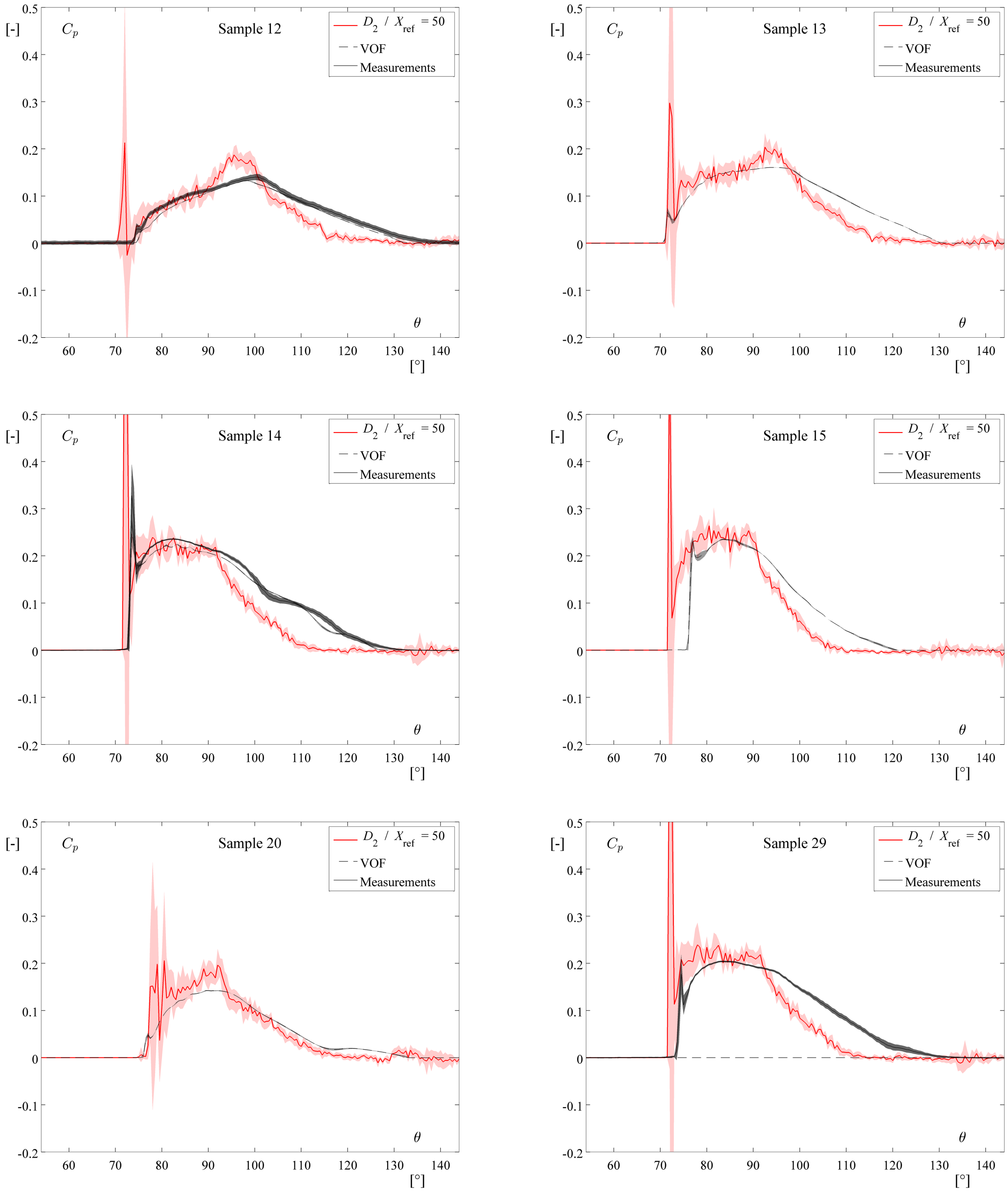


Figure 5.17 – Time history of C_p at the pressure sensors 12, 13, 14, 15, 20 and 29 on the buckets inner surface, which corresponds to the central region of the inner surface.

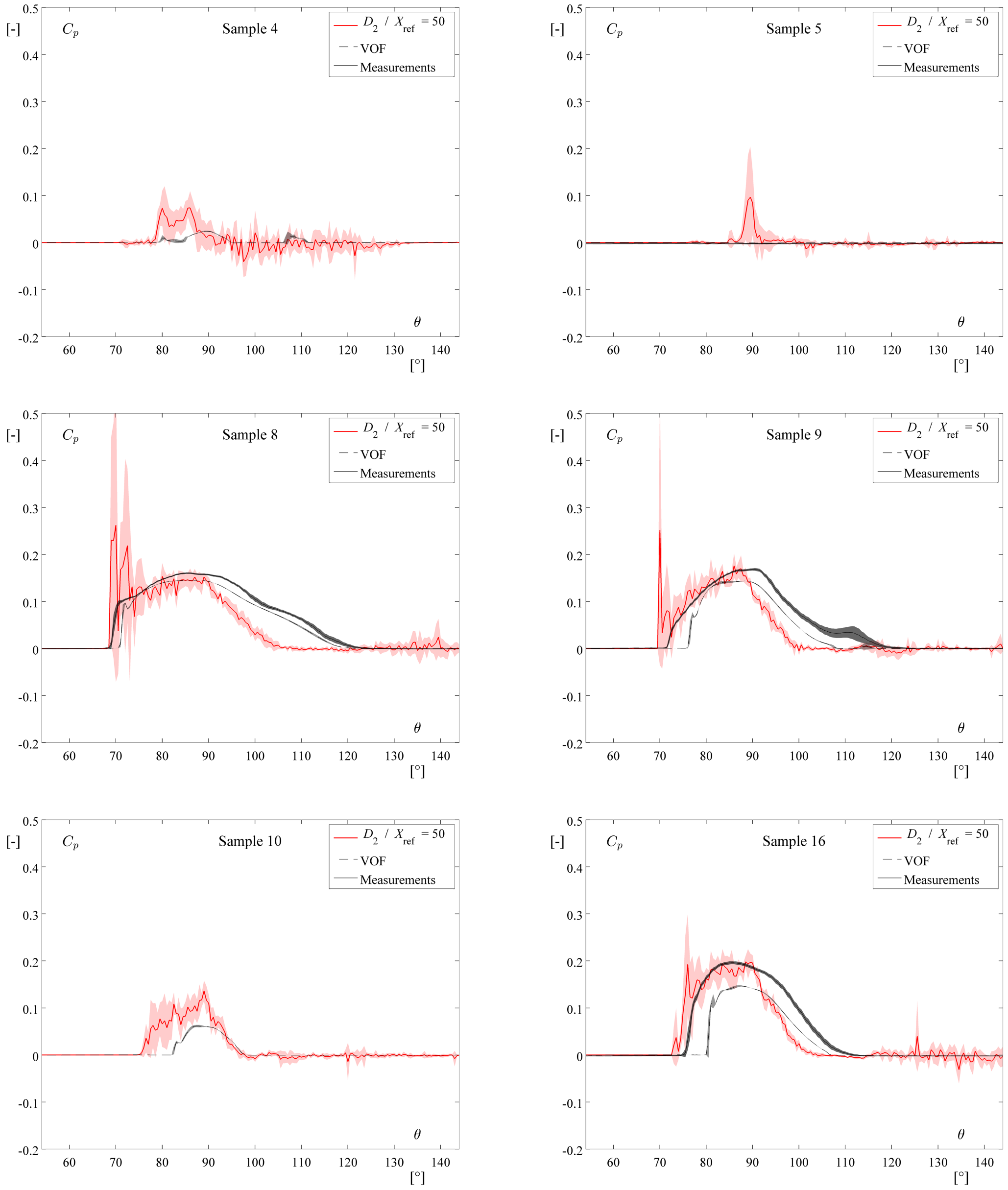


Figure 5.18 – Time history of C_p at the pressure sensors 4, 5, 8, 9, 10 and 16 on the buckets inner surface, which corresponds to the splitter side of the aft region of the inner surface.

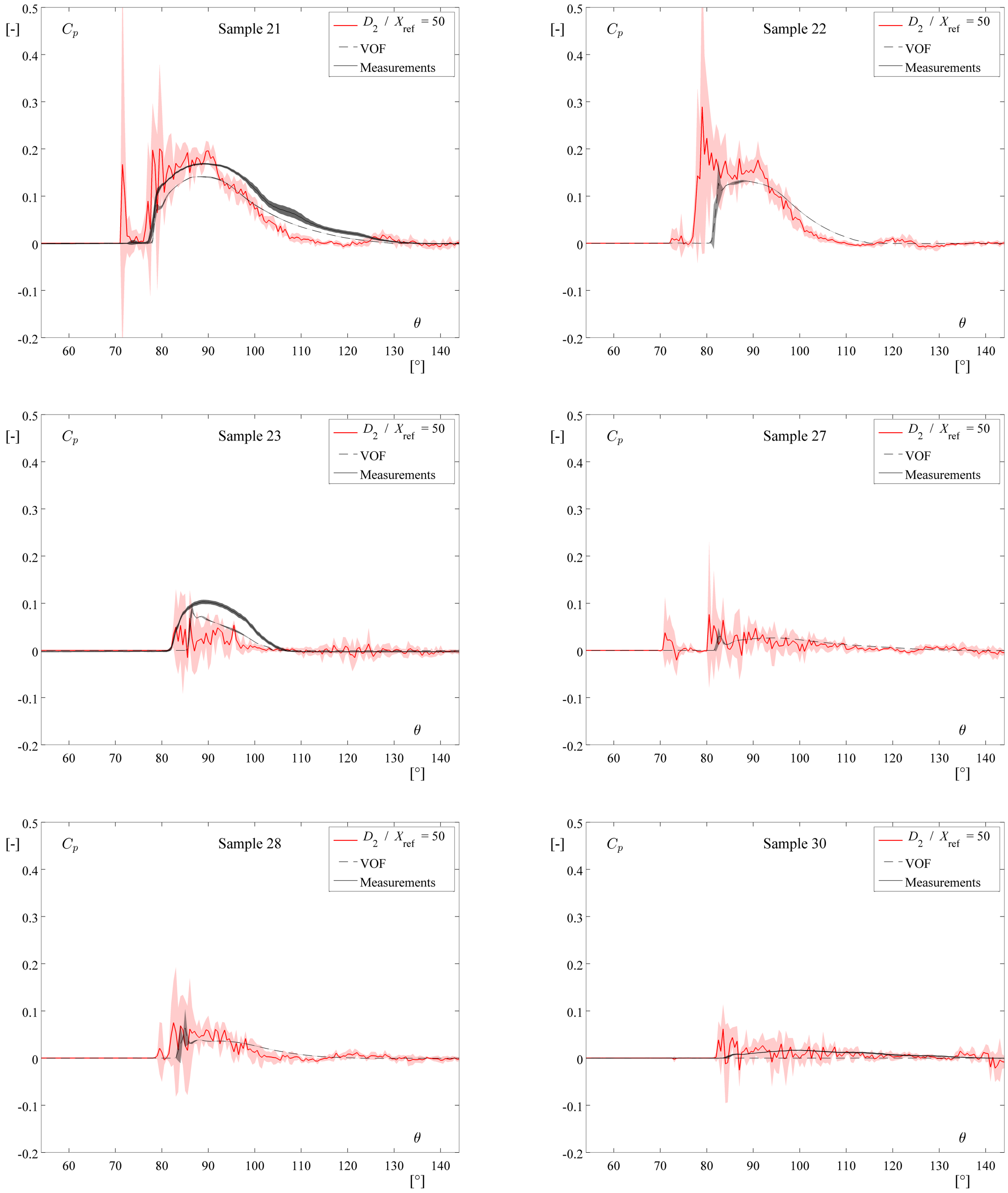


Figure 5.19 – Time history of C_p at the pressure sensors 21, 22, 23, 27, 28 and 30 on the buckets inner surface, which corresponds to the outer side of the aft region of the inner surface.

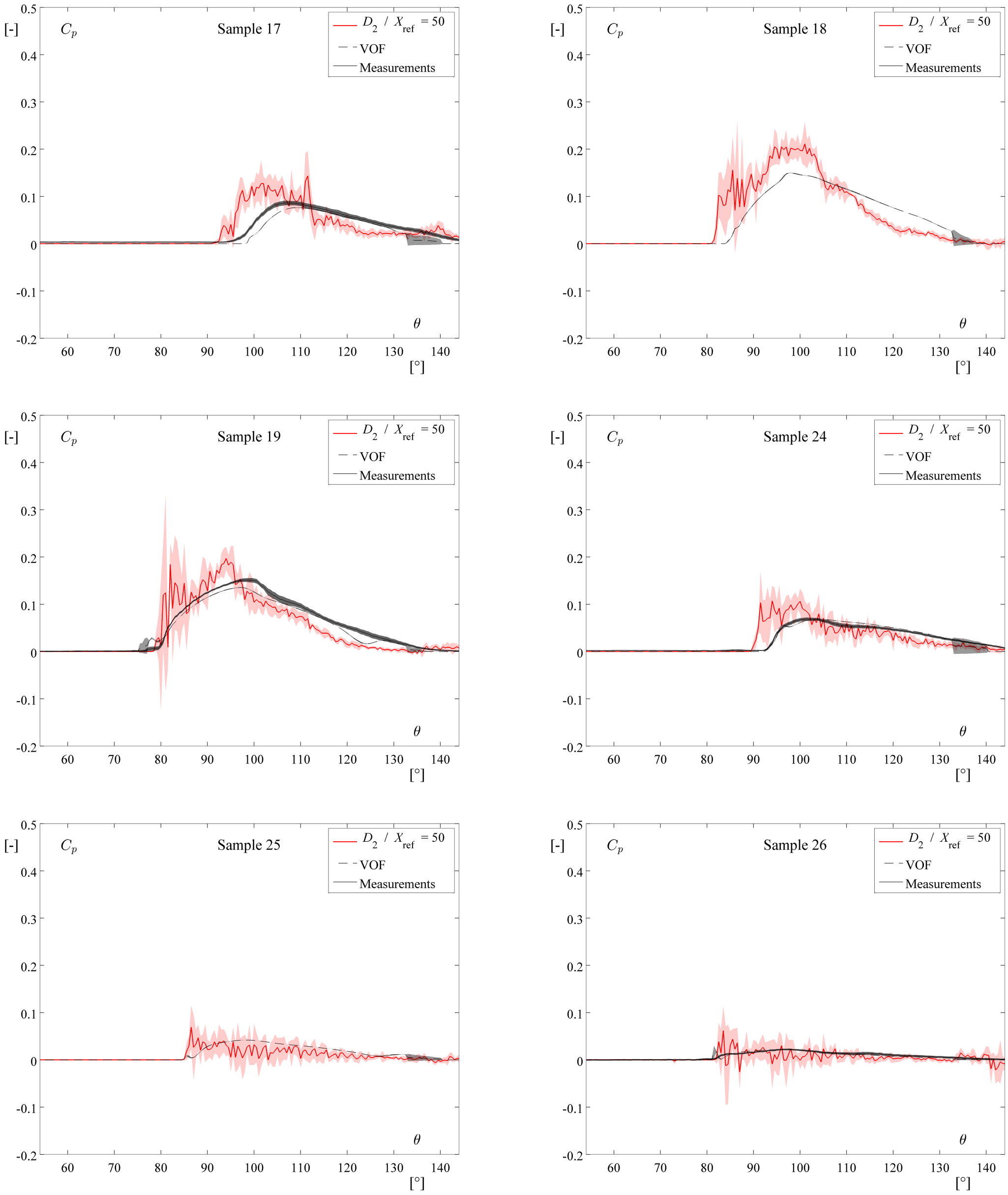


Figure 5.20 – Time history of C_p at the pressure sensors 17, 18, 19, 24, 25 and 26 on the buckets inner surface, which corresponds to the outer side of the frontal region of the inner surface.

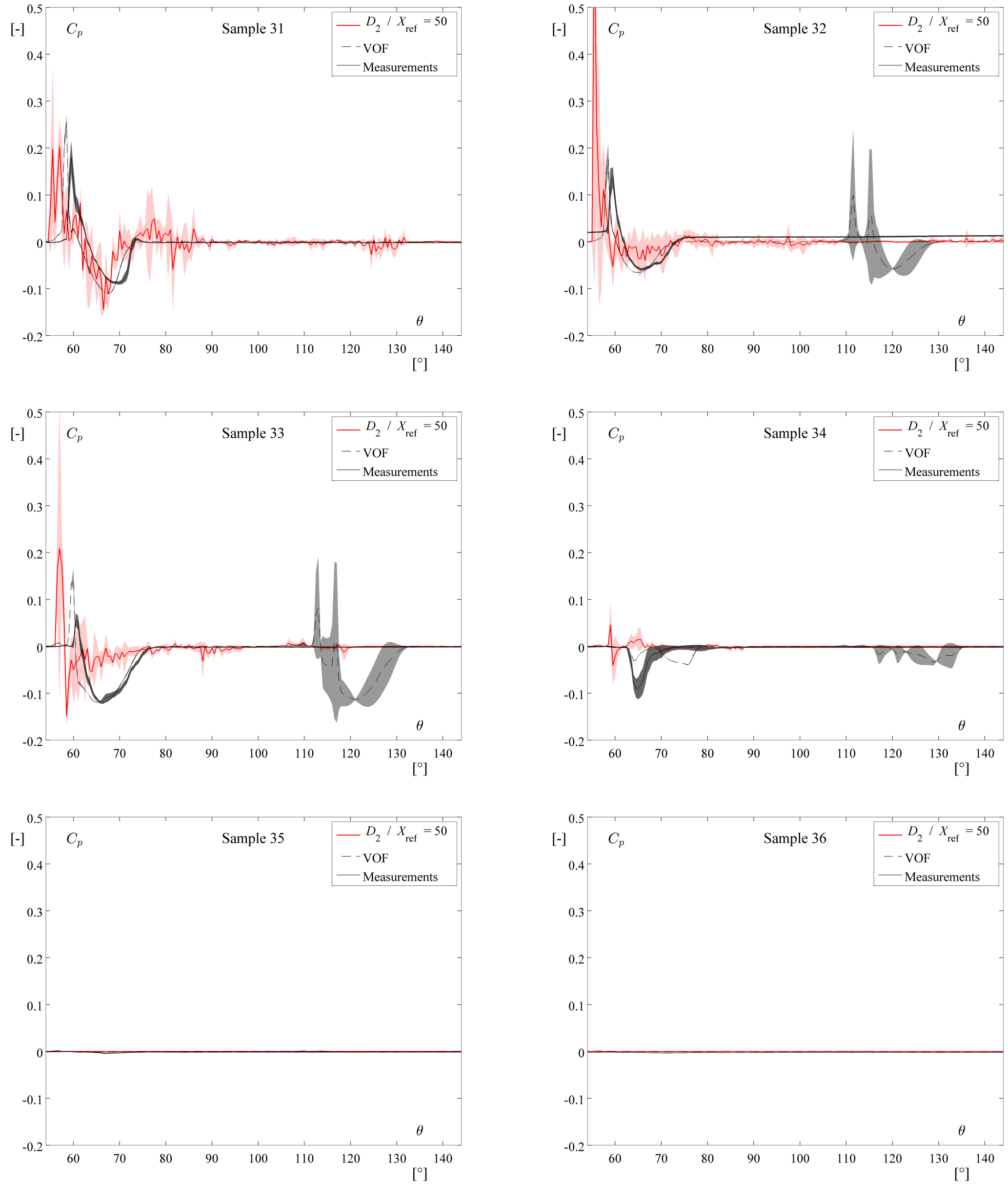


Figure 5.21 – Time history of C_p at the pressure sensors 31, 32, 33, 34, 35 and 36 on the buckets outer surface, which corresponds to the splitter edge of the outer surface.

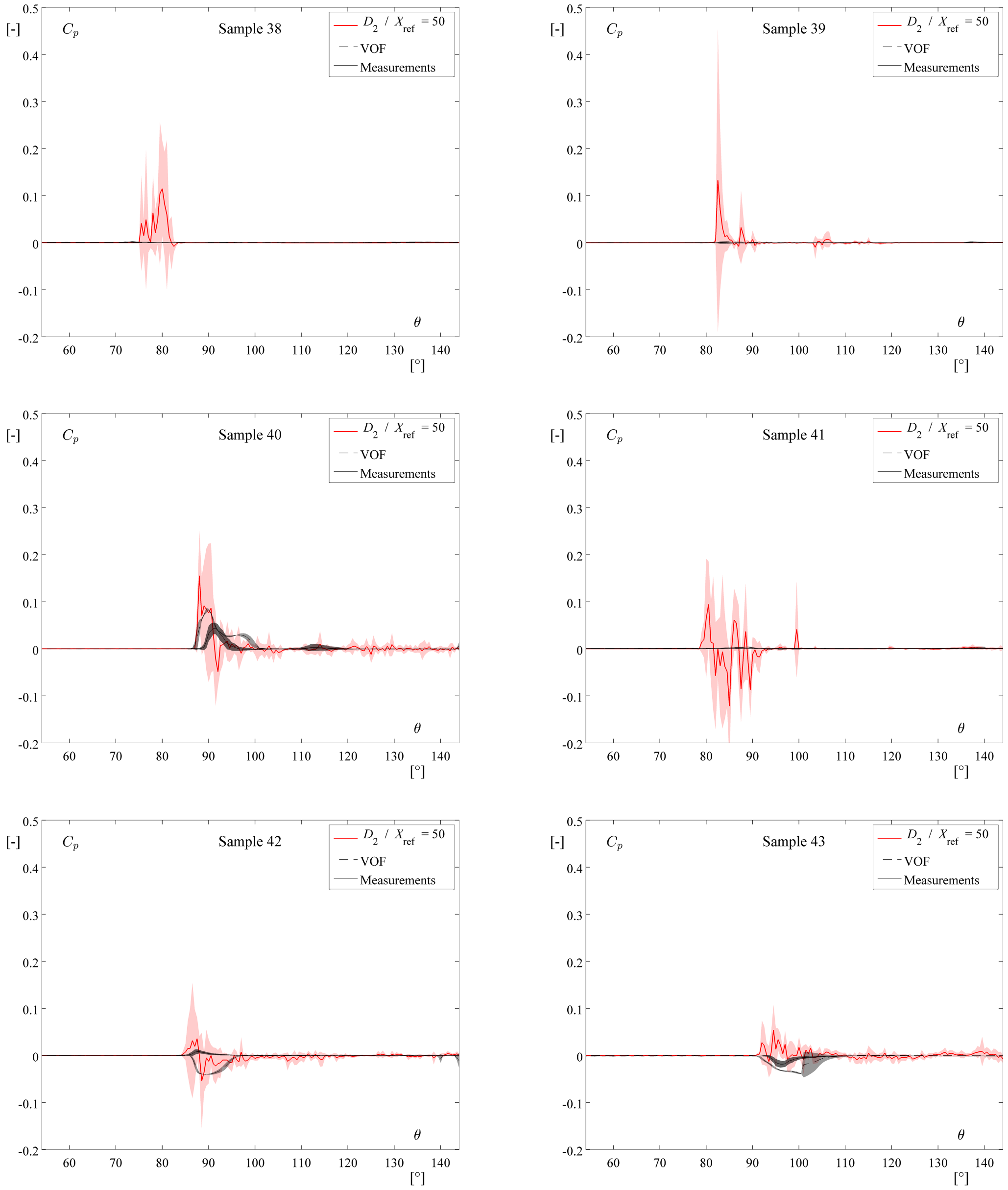


Figure 5.22 – Time history of C_p at the pressure sensors 38, 39, 40, 41, 42 and 43 on the buckets outer surface, which corresponds to the outer edge of the outer surface.

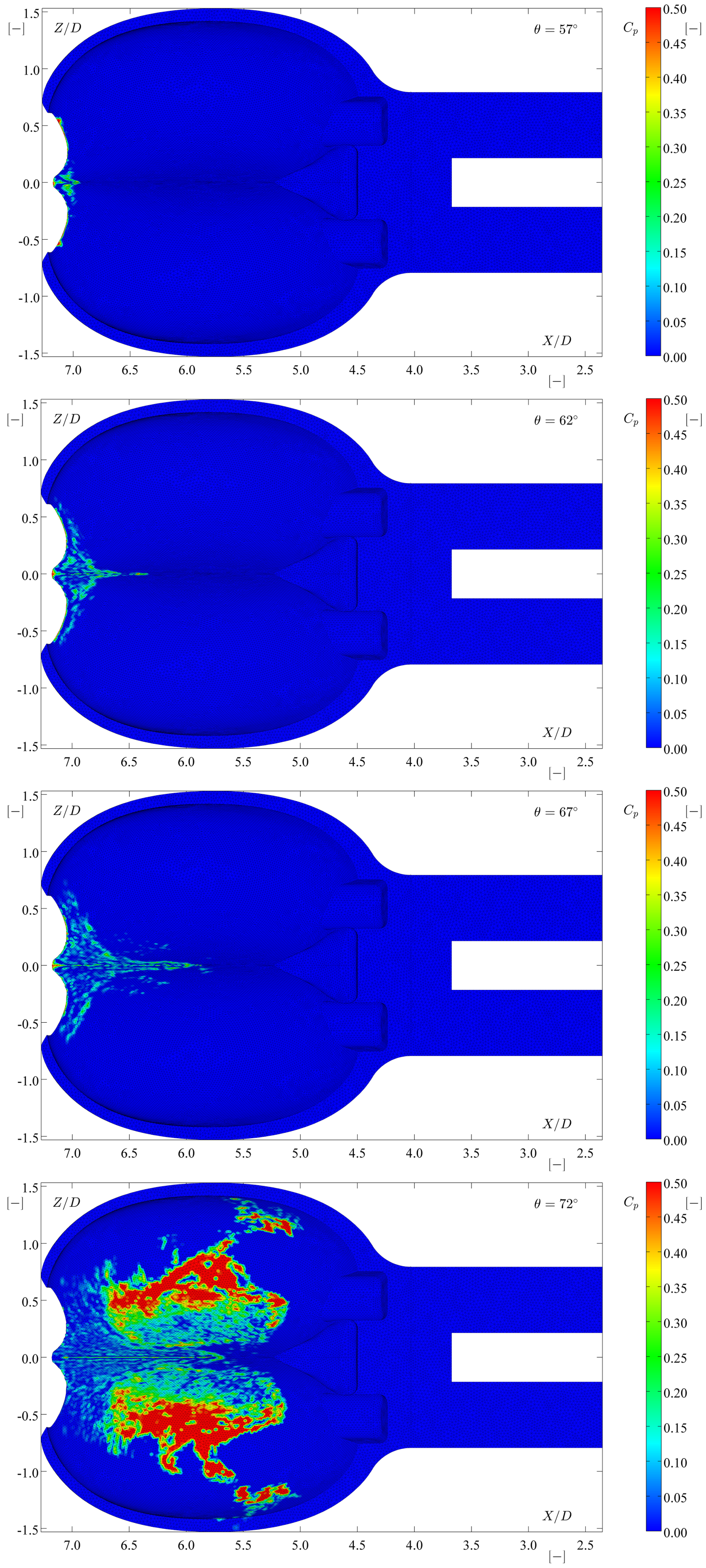


Figure 5.23 – Pressure field on the bucket inner wall for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°$ and $\theta = 72°$.

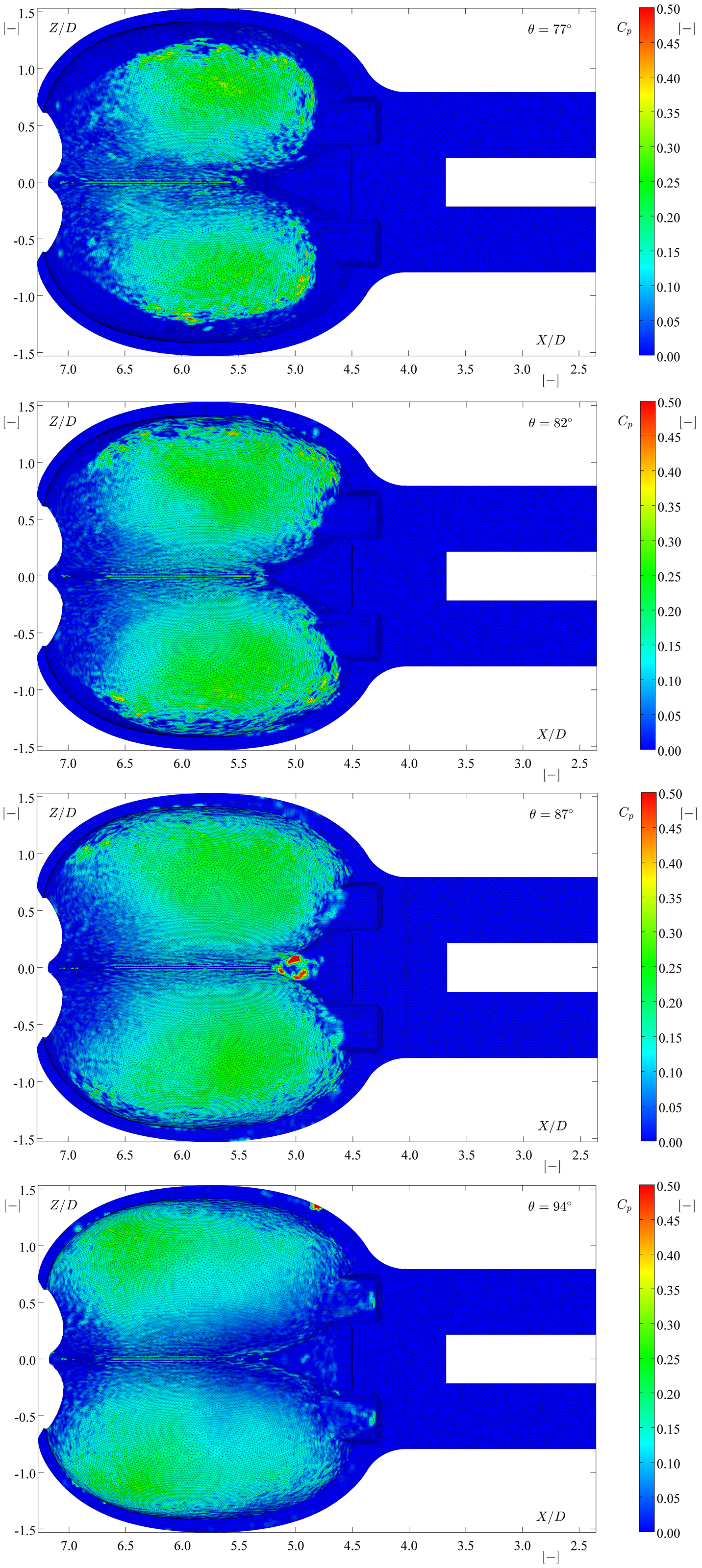


Figure 5.24 – Pressure field on the bucket inner wall for the impinging angles of $\theta = 77°, \theta = 82°, \theta = 87°$ and $\theta = 94°$.

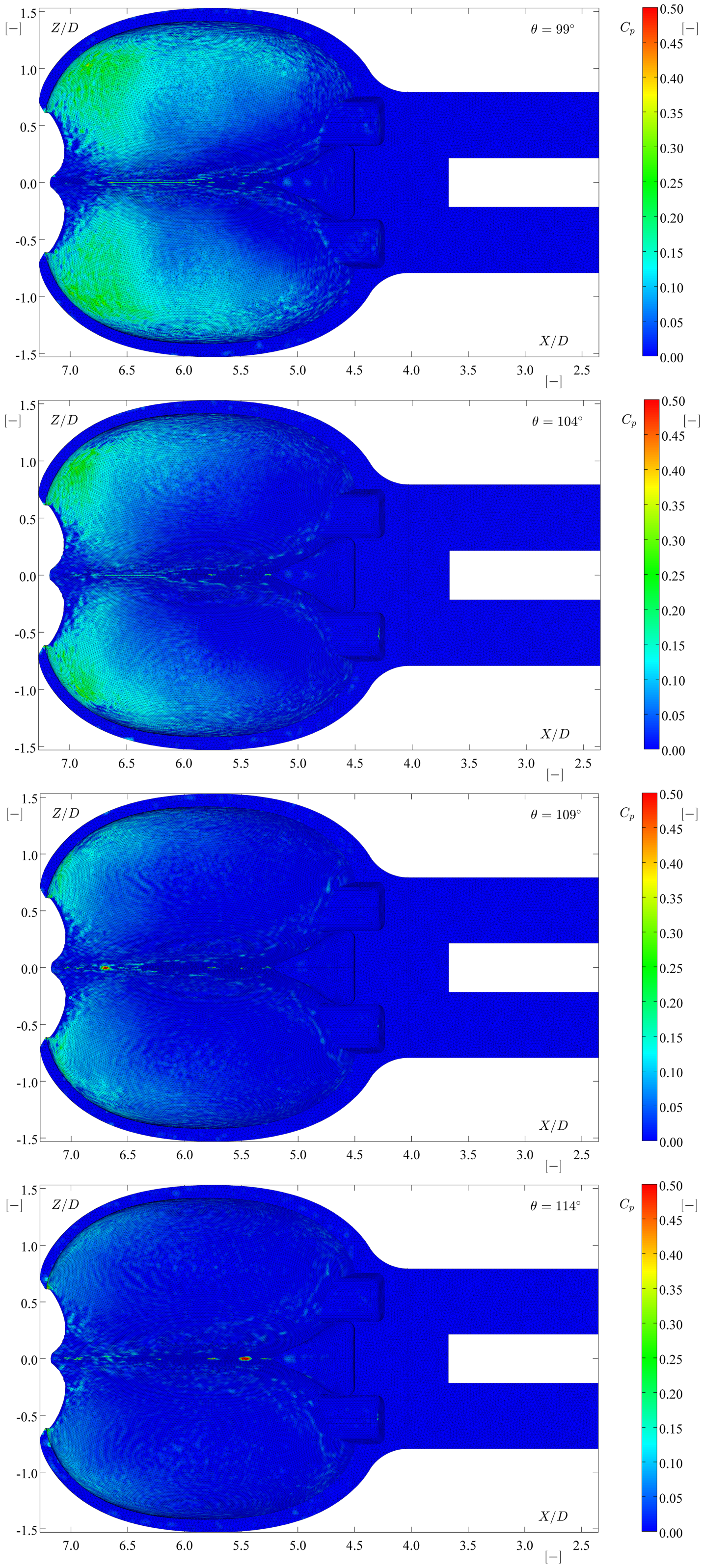


Figure 5.25 – Pressure field on the bucket inner wall for the impinging angles of $\theta = 99°, \theta = 104°, \theta = 109°$ and $\theta = 104°$.

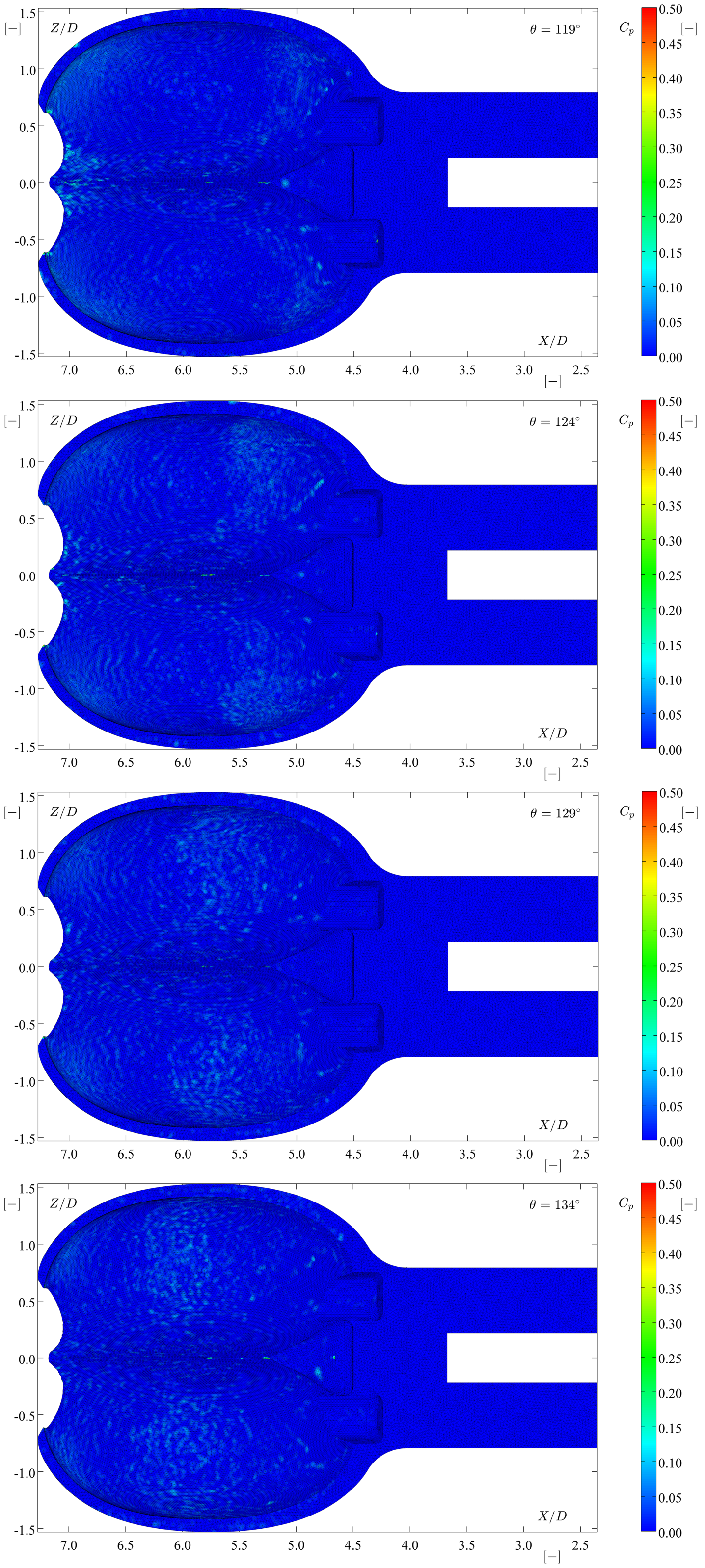


Figure 5.26 – Pressure field on the bucket inner wall for the impinging angles of $\theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$.

5.3 Free surface evolution analysis

5.3.1 Free surface reconstruction

The FVPM results contain the particles position and fluid properties, e.g. density, pressure and velocity. In order to reconstruct the free surface location, the algorithm of Zhu and Bridson [85] is implemented in a C++ code called *Particle2Grid*. First, an unstructured grid is generated in the vicinity of the free surface particles. The grid resolution is arbitrarily set as $X_{\text{ref}}/2$. Second, the distance to the free surface

$$\delta = |\boldsymbol{X} - \bar{\boldsymbol{X}}| - \bar{r} \tag{5.1}$$

is computed for each nodes of the grid featuring a position $\boldsymbol{X}$. The computation of δ depends on the weighted average of the nearby particles

$$\bar{\boldsymbol{X}} = \sum_{i}^{N} W_i^s \boldsymbol{X}_i \tag{5.2}$$

and weighted average of their radii

$$\bar{r} = \sum_{i}^{N} W_i^s r_i \tag{5.3}$$

where W_i^s is the weight defined as

$$W_i^s = \frac{W_i}{\sum_j W_j}. \tag{5.4}$$

The computation of this weight is performed using the Wendland kernel with a smoothing length $h = 1.3 \cdot X_{\text{ref}}$. This kernel is applied to each fluid particle i featuring a position $\boldsymbol{X}_i$ and a radius r_i. Finally, the rendering of the free surface is performed using *ParaView* and Python scripts. The unstructured grid containing the free surface distance is loaded in *ParaView*. A contour filter is applied to the grid data with the value $\delta = 0$ in order to represent the free surface. The Python script automatically performs this procedure according to the time step and defines the camera position, bucket angle, colors and opacity values.

5.3.2 Convergence study

The evolution of the free surface is presented in Figure 5.27 to 5.30 for two different discretizations: a coarse $D_2/X_{\text{ref}} = 20$ and a fine $D_2/X_{\text{ref}} = 40$. The bucket angles investigated correspond to the ones from the wall pressure field analysis given in the previous section. The free surface evolution is represented in the XZ view for the second bucket. In order to obtain a proper visualization, the water sheet is represented with a 0.5 opacity and the free surface located higher than the outer edge plane is discarded.

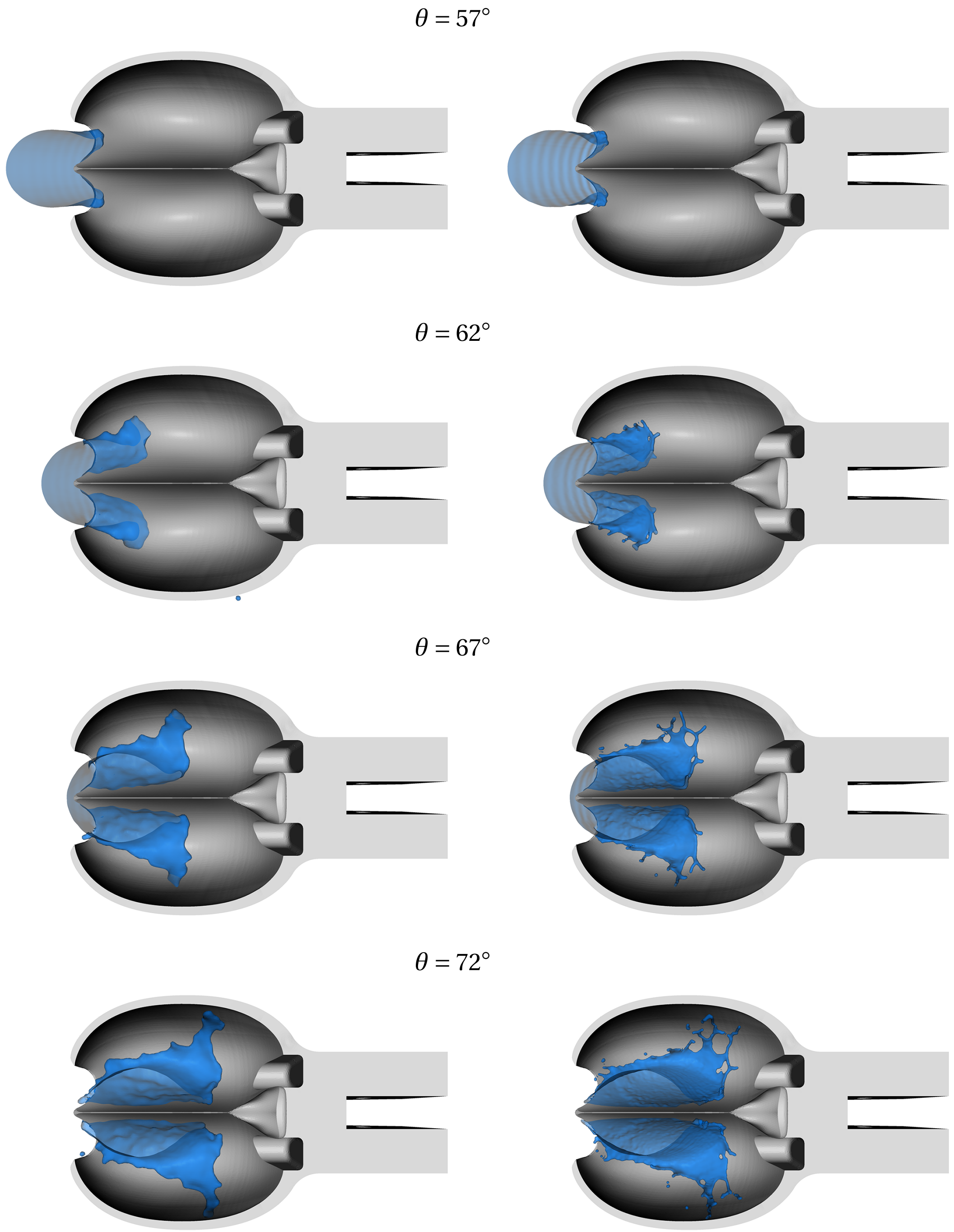


Figure 5.27 – Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°$ and $\theta = 72°$. The spatial discretization investigated are $D_2 / X_{\mathrm{ref}} = 20$ (left) and $D_2 / X_{\mathrm{ref}} = 40$ (right).

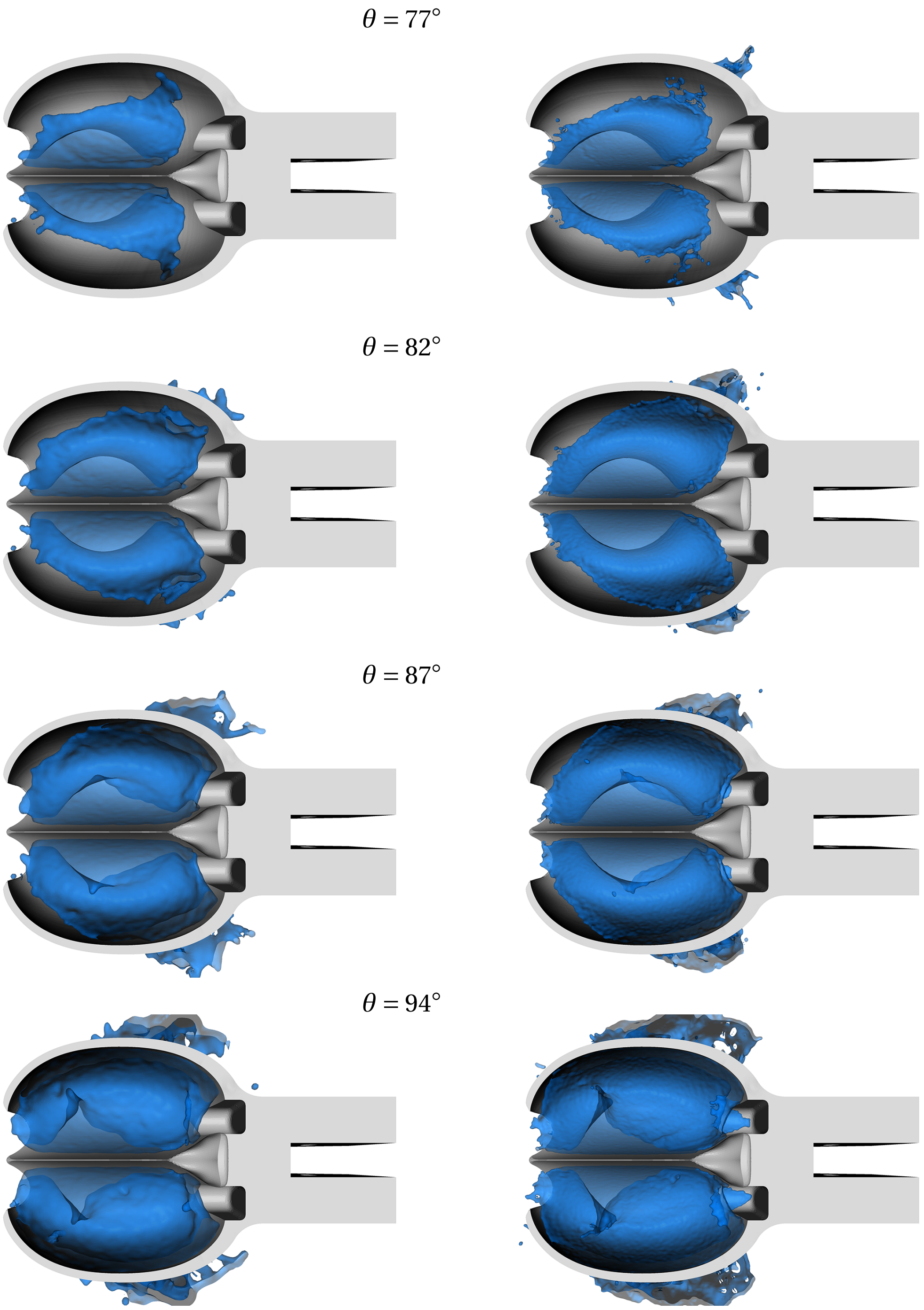


Figure 5.28 – Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 77°, \theta = 82°, \theta = 87°$ and $\theta = 94°$. The spatial discretization investigated are $D_2/X_{\text{ref}} = 20$ (left) and $D_2/X_{\text{ref}} = 40$ (right).

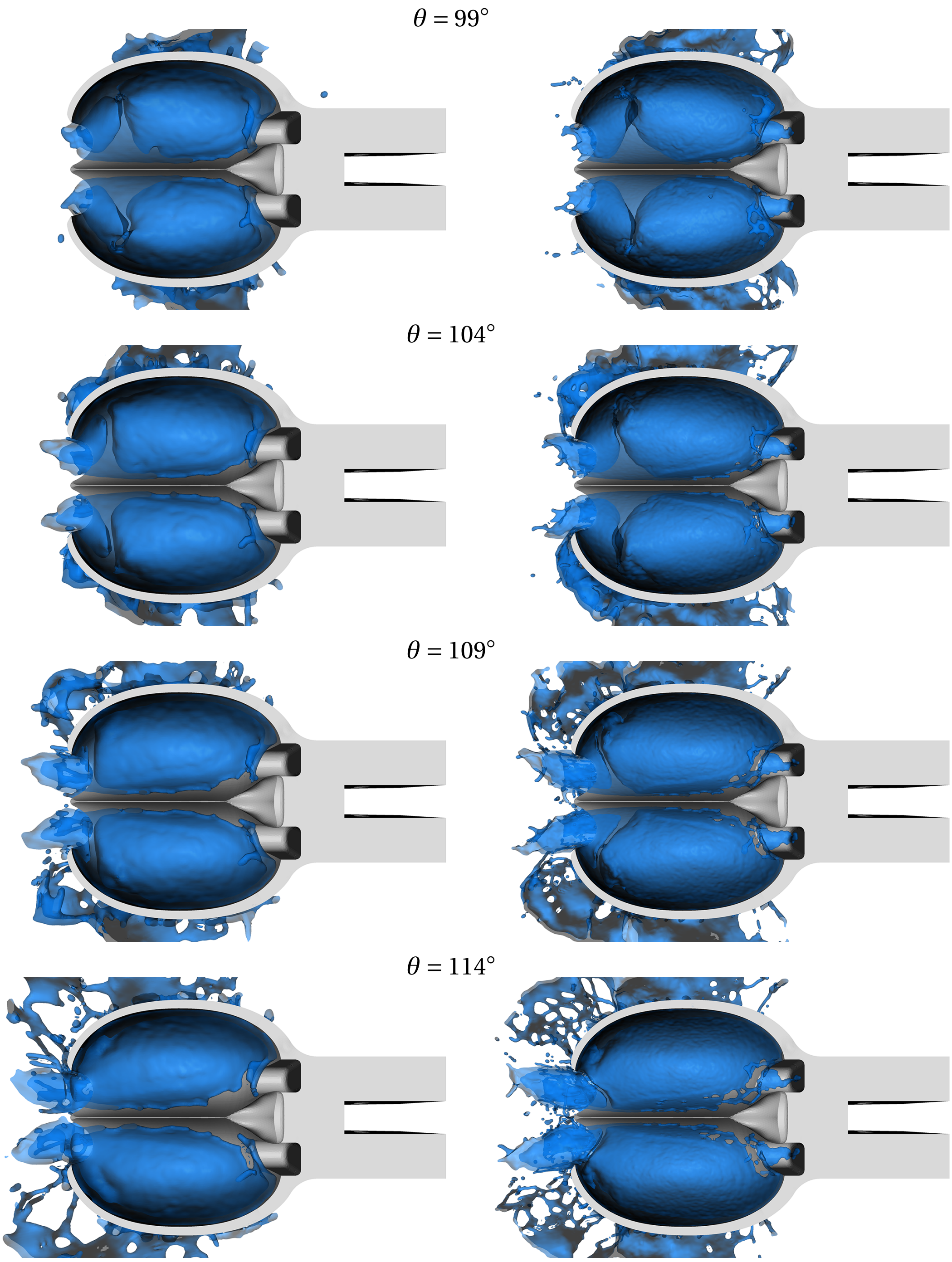


Figure 5.29 – Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 99°, \theta = 104°, \theta = 109°$ and $\theta = 114°$. The spatial discretization investigated are $D_2/X_{\mathrm{ref}} = 20$ (left) and $D_2/X_{\mathrm{ref}} = 40$ (right).

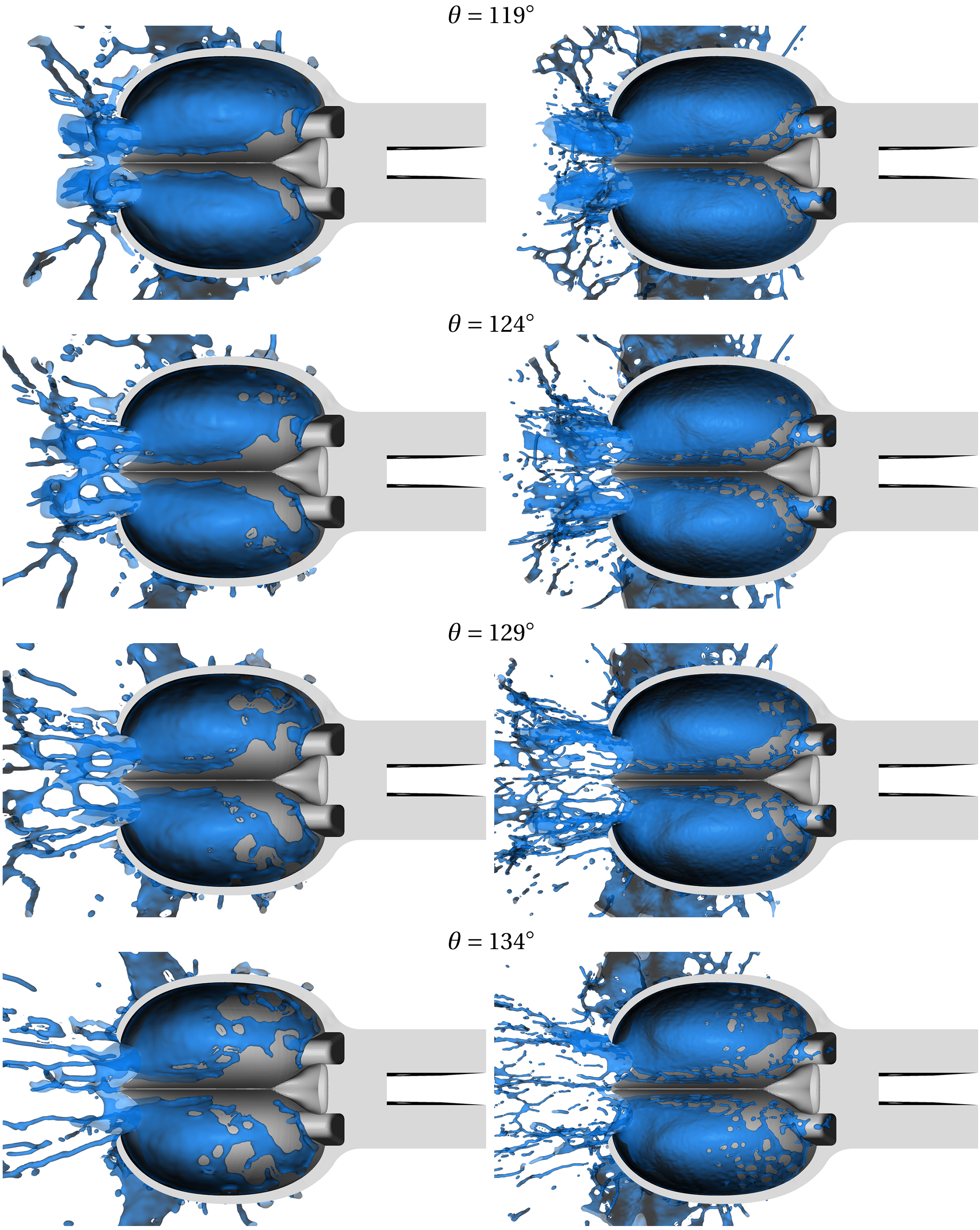


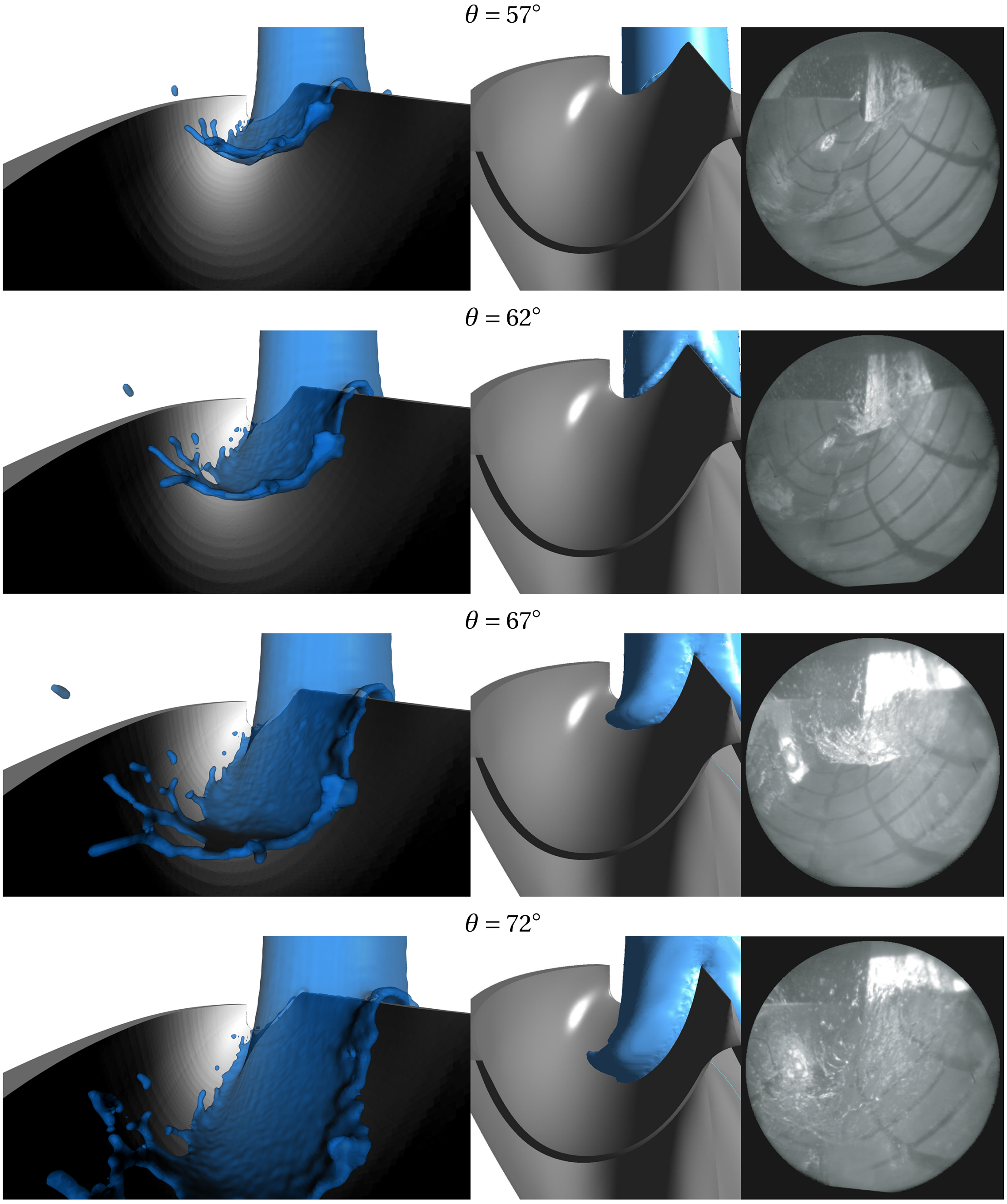
Figure 5.30 – Influence of the particles resolution on the free surface evolution for the impinging angles of $\theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$. The spatial discretization investigated are $D_2/X_{\text{ref}} = 20$ (left) and $D_2/X_{\text{ref}} = 40$ (right).

The cut of the water jet by the outer edge plane is well represented for both discretization. At the impingement first stage, i.e. $\theta = 57°$ and $\theta = 62°$, pressure waves are generated near the cutout and reflected in direction of the jet inlet. This phenomenon is due to the flow compressibility and has a bigger influence for the fine discretization. The disturbance of the jet since $\theta = 94°$ is due to the next bucket, i.e bucket 3 and is also well captured for both resolutions. The evolution of the wet surface is similar for both discretization but the fine resolution presents the advantage to capture more details of the flow. The outflow of the previous bucket, i.e bucket 1, features some delay with the coarse discretization at $\theta = 77°$ and $\theta = 82°$. Since $\theta = 99°$ the flow exits from the cutout. Both resolutions capture this behavior but more details are highlighted with the fine discretization. The water sheet evolution is overall well reproduced for both resolutions. However, the fine discretization is better suited in the case of a full runner simulation with casing interaction. Indeed, the fine resolution predicts this diverging flow with more accuracy because the size of the water droplets is smaller. An accurate prediction of the water sheets trajectories is important to prevent the apparition of the heeling phenomenon, i.e. the decrease of performance due to the impingement of the water jet on the next bucket. Moreover, the prediction of the water sheet trajectories is also important to check if any bad interactions with the casing take place.

5.3.3 Results

The relative flow pattern inside the second bucket is compared to the VOF result and flow high-speed visualization from Perrig [65] in Figure 5.31 to 5.34. The point of view is taken from the high-speed visualization. The FVPM free surface results feature a spatial discretization which corresponds to $D_2 / X_{\text{ref}} = 40$ and a 0.5 opacity is applied to the free surface visualization.

Between $\theta = 57°$ and $\theta = 67°$, the water jet starts to impinge the buckets and a small disturbance of the water sheet is highlighted in experimental and FVPM results. At $\theta = 72°$, the water sheet impinges the central zone of the buckets inner wall. This phenomenon, called water-hammer by Perrig [65], is well reproduced by FVPM. Moreover, the FVPM results are similar to the flow visualization. The FVPM water sheet, which is disturbed by the impingement first stage, impinges on the buckets central zone and produces the pressure peak highlighted in Figure 5.17. The VOF water sheet remains more compact, which explains the difficulty to capture the pressure peak and the smoother pressure profile. Between $\theta = 87°$ and $\theta = 99°$, the water jet is perturbed by the next bucket, i.e. bucket 3 and the FVPM water sheet spreads more on the inner wall compared to VOF. At $\theta = 104°$, the last portion of the jet arrives on splitter while the water sheet starts to exit the bucket by the cutout, which is confirmed by the pressure profile of sensor 17 in Figure 5.20. Between $\theta = 109°$ and $\theta = 134°$, the remaining water is evacuated from the bucket. The VOF results are in advance compared to the measurements and FVPM results.



$\theta = 57°$

$\theta = 62°$

$\theta = 67°$

$\theta = 72°$

Figure 5.31 – Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 57°, \theta = 62°, \theta = 67°$ and $\theta = 72°$.

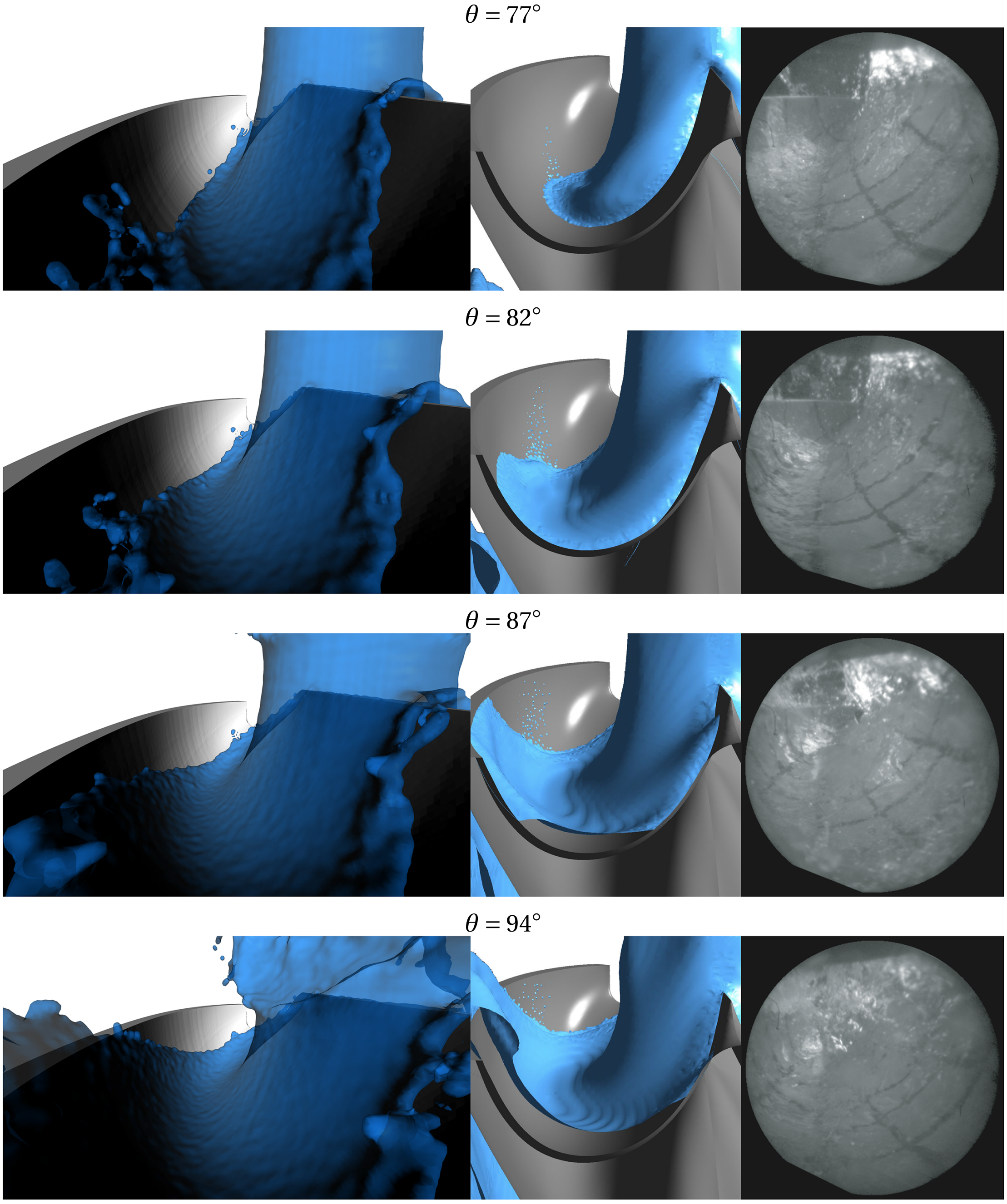
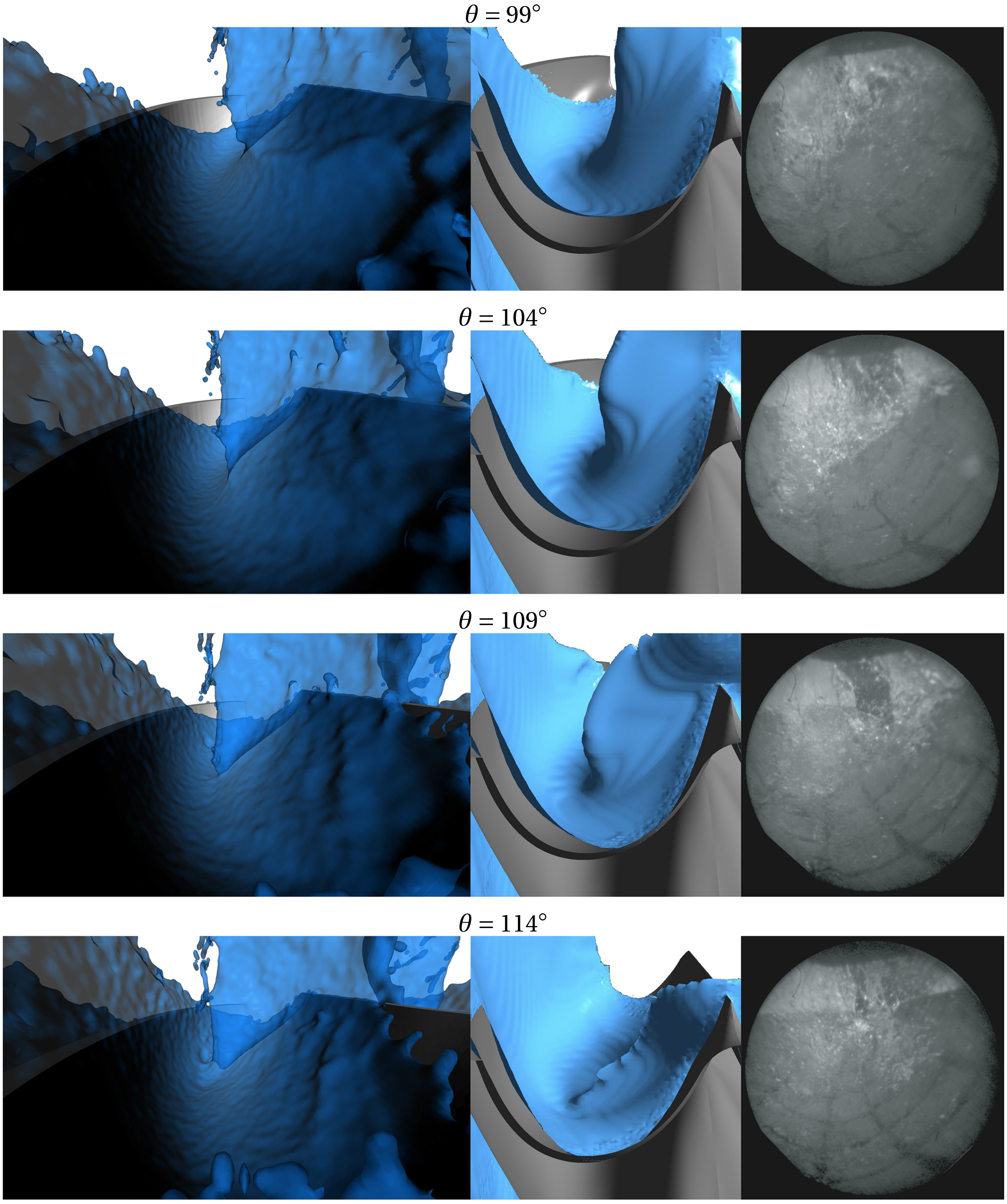


Figure 5.32 – Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 77°, \theta = 82°, \theta = 87°$ and $\theta = 94°$.



$\theta = 99°$

$\theta = 104°$

$\theta = 109°$

$\theta = 114°$

Figure 5.33 – Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 99°, \theta = 104°, \theta = 109°$ and $\theta = 114°$.

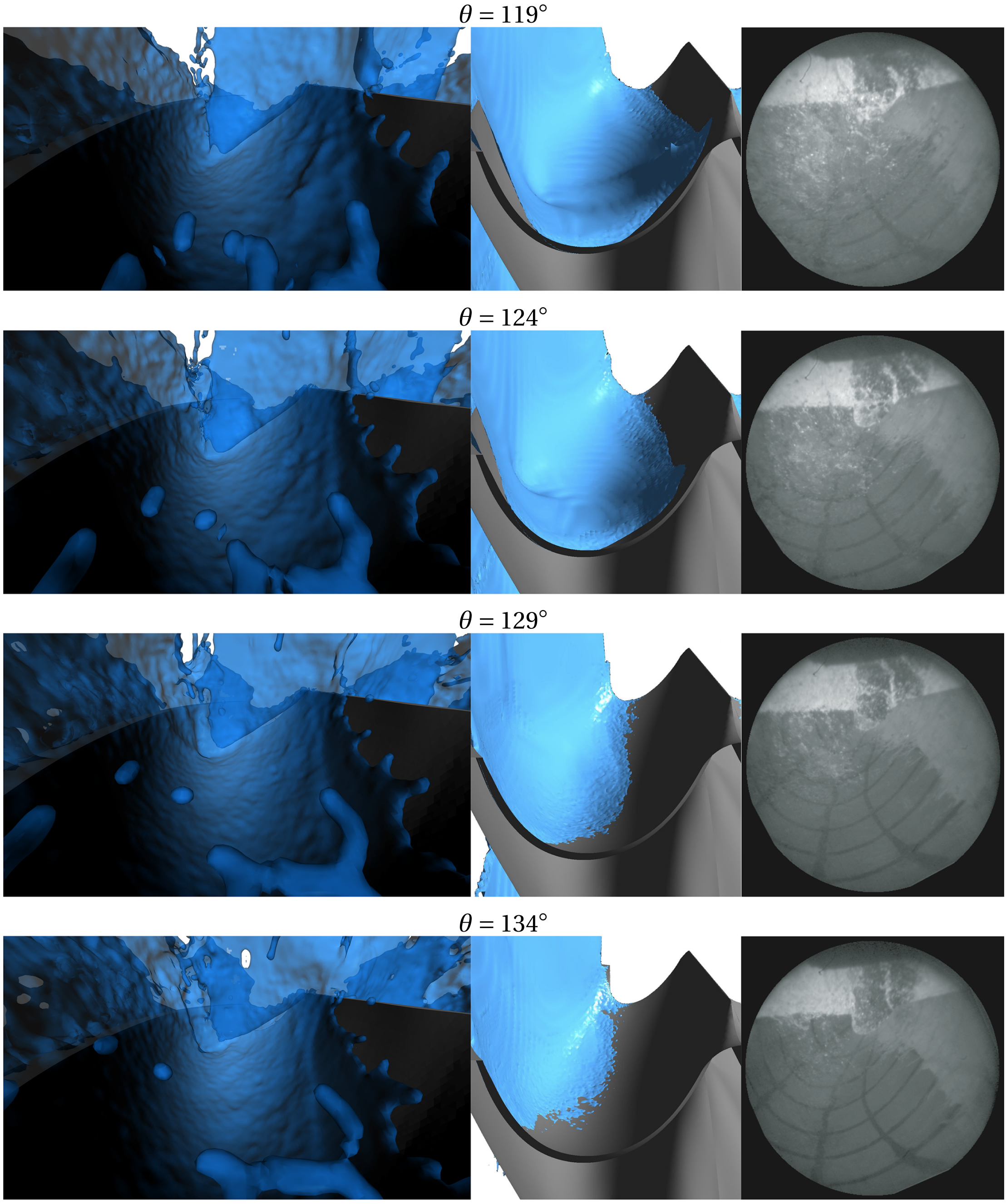


Figure 5.34 – Comparison of the relative flow pattern inside the buckets between FVPM (right), VOF (middle) and experimental (right) for the impinging angles of $\theta = 119°, \theta = 124°, \theta = 129°$ and $\theta = 134°$.

6 Conclusions and Perspectives

6.1 Conclusions

Numerical simulations of free jet deviations are performed using particle-based methods and validated by conventional grid-based simulations and experimental data. In order to decrease the computational time required for the particle-based simulation, the computations are massively parallelized and run on two different clusters: an IBM blade center and the *Lemanicus* BG/Q. The present study highlights the efficiency of the adaptive domain decomposition process to dynamically balance the load in the subdomains. The orbital shaking and Pelton buckets test cases show a decrease of the computational time by a factor between three and four. Moreover, the proposed ADD strategy is well suited for complex free surface applications and highly dispersed particles system. The computing cost of the ADD process remains reasonable, i.e. about one tenth of the solver iteration time.

The FPM flow simulations of the impinging jet on a flat plate are validated with the available VOF computations and measurements from Kvicinsky et al. [41]. The new shifting strategy as well as the improved boundary treatment are validated and implemented within *SPHEROS*. The FPM solver allows to represent properly the physic of free jet deviation. The free surface location is in good agreement with the measurements and VOF results for the three impinging angles and all the spatial discretizations investigated. Therefore, the use of particle based method to obtain animations of free surface flows is fast and efficient. However, the FPM results highlight that a fine spatial discretization is required to capture accurately the pressure profile, especially when dealing with high pressure gradients. The FVPM simulations are compared to the FPM results. The FVPM results fit better the VOF and measurements but the computing time is increased by a factor two. Nevertheless, the use of FVPM is mandatory to simulate the flow in Pelton buckets. Indeed, the solid boundary treatment in FVPM is simpler and more accurate for complex geometries compared to the solid boundary treatment in FPM.

The FVPM flow simulations of a steady bucket at different impinging angles are validated with VOF computations and measurements from Kvicinsky et al. [39]. It is demonstrated that the FVPM ensures the convergence of the results thanks to its conservative and consistent formulation. Indeed, the more particles are added, the better are the results. Moreover, if the CFL condition is fulfilled, no instabilities are observed with the increase of the spatial resolution. The wall pressure field fits qualitatively well the VOF computations and measurements for all the impinging angles investigated. The FVPM wall pressure field features more numerical fluctuations compared to VOF. However, these fluctuations decrease when the spatial resolution is increased.

The FVPM flow simulation of rotating Pelton buckets are validated with VOF computations and measurements from Perrig [65]. The convergence of the method according to the spatial discretization is demonstrated by the convergence of the resulting torque evolution. The pressure field in the buckets inner wall is in good agreement with the experimental and numerical data. However, the numerical fluctuations of the FVPM signals are always larger than the RMS fluctuations of the VOF results. These numerical fluctuations are partly explained by the lack of particles close to the sampling location, e.g. the buckets outer wall. Moreover, FVPM is better suited to capture the pressure peak, which occurs at $\theta = 72°$ in the central zone of the buckets inner wall. This phenomenon, called waterhammer by Perrig [65], is due to the compressibility of the water/air mixture, which impinges on the central zone at $\theta = 72°$. This mixture is generated by the impingement first stage at the buckets cutout. Therefore, the weakly compressible approach of FVPM is better suited to capture this phenomenon than the incompressible approach of VOF. The relative flow pattern in the buckets is in good agreement with the flow high-speed visualization for all the spatial resolutions investigated. However, the fine discretization provides more accuracy to represent the water sheets because the size of the water droplets is smaller. Consequently, the FVPM simulation is an accurate tool to capture the deviation of high-speed water jets by rotating Pelton buckets. A spatial discretization of $D_2/X_{\text{ref}} = 40$ is recommended to obtain a converged simulation with a good representation of the water sheets

6.2 Perspectives

In the present research project, the deviation of a free jet by rotating Pelton buckets is simulated using FVPM. The numerical fluctuation of the wall pressure field is significant due to the lack of particles close to the sampling location. Therefore, the use of particle splitting should be investigated to mitigate this phenomenon. As the flow is diverging, the particle splitting is also a good option to improve the accuracy of the simulation in the water sheets. Another option that could be investigated to improve the wall pressure field is the use of the ALE formulation to have more particles which remain close to the buckets geometry all along the simulation. In order to capture accurately the magnitude of the pressure peak at $\theta = 72°$, the possibility to adapt the sound speed according to the location in the flow should be investigated.

The computing costs of the FVPM simulations remain significant with the today's computing resources. However, the performance of *SPHEROS* could be improved to obtain a better peak performance efficiency on the *Lemanicus* BG/Q. Therefore, a code profiling could be applied to identify the algorithms which have to be optimized to improve the code performance. Moreover, the possibility to adapt some parts of the code to run on GPU instead of CPU should be investigated to allow large computations without using a BG/Q.

Finally, the simulation of the flow in rotating Pelton buckets using *SPHEROS* could be used to optimize the buckets geometry according to given operating conditions. A genetic algorithm may be coupled to the *SPHEROS* simulations in order to shape the design of the buckets wall. The basis of this optimization process is the parametric representation of the buckets geometry given in appendix A.

A Parametric Model of a Pelton Bucket

Introduction

In order to run CFD-based optimization of Pelton runners, a parametric model of a bucket is proposed by Andolfatto [4]. The inner bucket wall is defined by 4 Bézier surfaces. Some general dimensions of the bucket are assumed to be predefined according to global hydraulic characteristics of the machine to be designed. An additional set of parameters is proposed to completely define the bucket inner wall. The bucket outer wall is defined by an offset of the inner wall using a specified thickness map. Finally, all the parameters used are inventoried and classified to generate the geometry, which is given as an input of the numerical simulation.

Definition of the inner surface

General dimensions

Some general dimensions of the Pelton bucket are considered to be fixed according to previous studies. These dimensions are represented in Figure A.1.

The intersection of the water jet axis with the outer edge plane of the bucket is named O and its coordinates are $(0,0,0)$ in the outline given in Figure A.2. As the bucket is assumed to be symmetric with respect to the XY plane, only the half bucket with positive coordinates along the Z axis will be described.

Definition of physical points

A set of physical points, with specific characteristics are defined on the inner surface:

- T is the tip of the bucket, with the coordinate $(C_2, -Y_t, 0)$;
- Ce is the other extremity of the cutout, with the coordinate $(A_2, 0, E/2)$;

- Cb is the bottom point of the cutout edge, where the edge tangent is normal to the Y axis, with the coordinate $(X_{Cb},-Y_{Cb},Z_{Cb})$;
- Bt is the bottom of the inner surface where the surface normal is oriented along Y, with the coordinate $(X_{Bt},-F,B/4)$;
- Om is the extreme point of the outlet edge along the Z direction, with the coordinate $(X_{Bt},0,B/2)$;
- Sm is the intermediate point of the inlet edge, with the coordinate $(X_{Bt},Y_{Sm},0)$;
- Se is the extreme point of the inlet edge, with the coordinate $(-C_3,Y_{Se},0)$;
- Oe is the extreme point of the outlet edge along the X direction, with the coordinate $(-A_1,0,B/4)$.

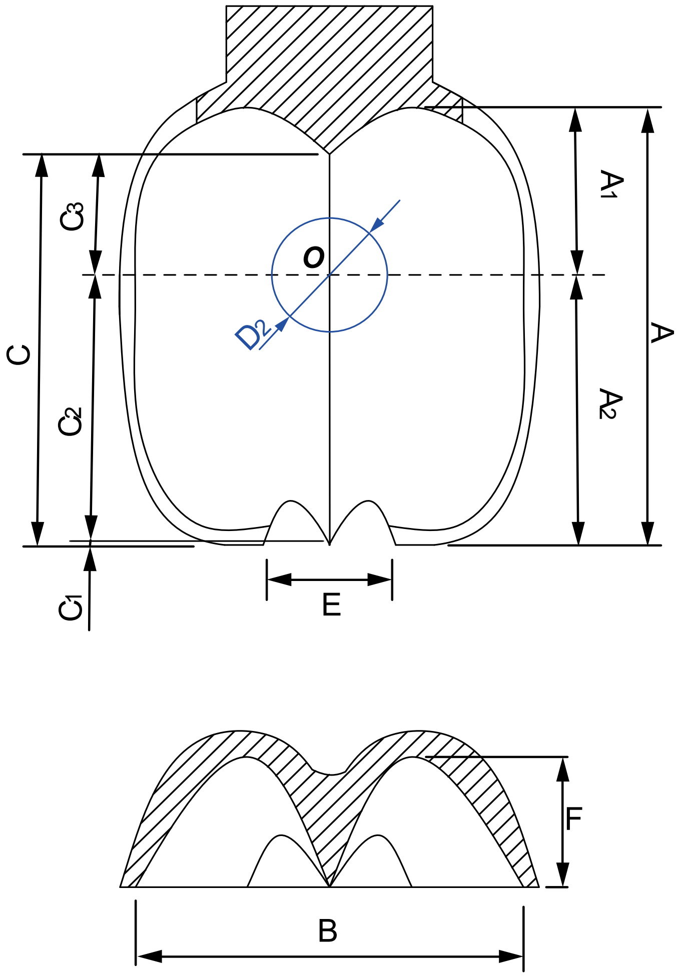


Figure A.1 – General dimensions defined in [4].

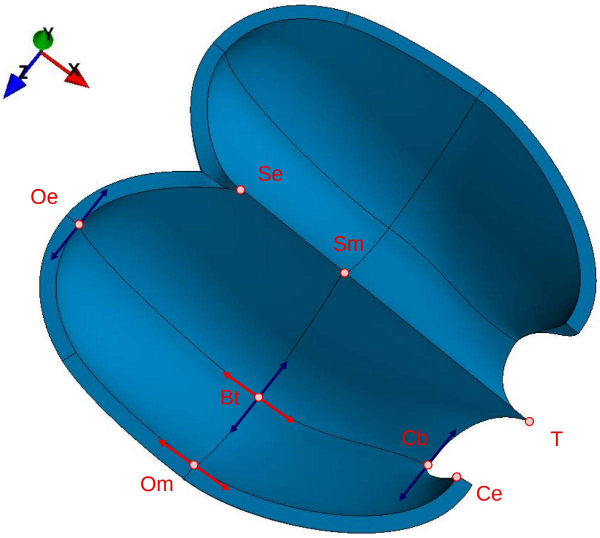


Figure A.2 – Physical points considered to describe the inner surface [4].

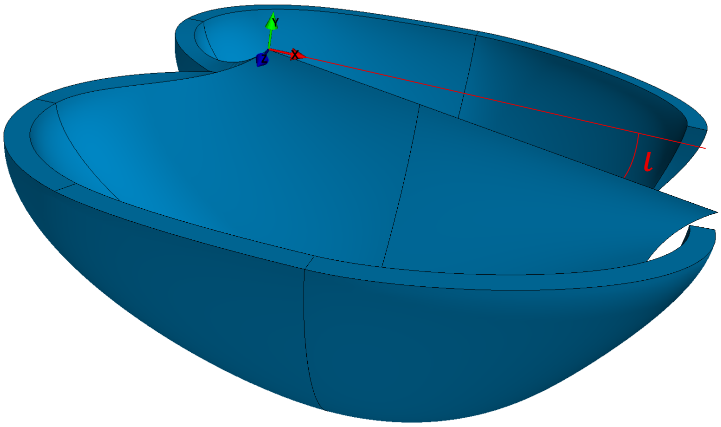


Figure A.3 – Definition of the inlet orientation angle ι [4].

The inlet orientation angle ι is defined between the inlet edge and the X direction, as depicted in Figure A.3. Then the coordinates of the physical points Sm and Se along the Y direction can be expressed with the following expressions:

$$Y_{Sm} = C_2 \cdot \tan\iota - Y_T \tag{A.1}$$

$$Y_{Se} = C \cdot \tan\iota - Y_T \tag{A.2}$$

The definition of the physical point finally requires the five additional parameters on top of the general dimensions previously defined: ι, Y_T, X_{Bt}, X_{Cb}, Y_{Cb}, Z_{Cb}. Four Bézier surfaces of order 4 × 4 are defined based on these physical points. A C^1 continuity between the four surfaces is ensured by imposing a symmetry of the four control points at each vertex shared by several surfaces.

Definition of the Bézier surfaces

First surface

The control points of the first surface are built according to the scheme described in Figure A.4. This surface depends on the following parameters:

- the inlet angle β_1 defines the angle between the inlet surface and the XY plane;
- the cutout tip angle α_T defines the orientation of the cutout edge;
- the cutout inlet angle $\beta_{1,Cb}$ defines the angle between the cutout inlet surface and the YZ plane;
- the cutout rotation angle α_{Cb} defines the local rotation of the cutout edge around Y;
- the $\lambda_\bullet$ that are fixed ratio of the general dimensions for the four surfaces.

Second surface

The C^1 continuity between the first and the second surface directly defines 8 of the 16 control points of the second surfaces, as shown in Figure A.5. The other part of the inlet surface between Sm and Se is oriented with the same β_1 angle with respect to the XY plane.

The outlet surface in Oe is oriented with an angle $\beta_{\overline{1},Oe}$ around Z. The angle α_{Se} is arbitrarily set to 45° in order to create a smooth edge between Se and Oe but this edge is unlikely to have an influence on the hydrodynamic behaviour of the bucket.

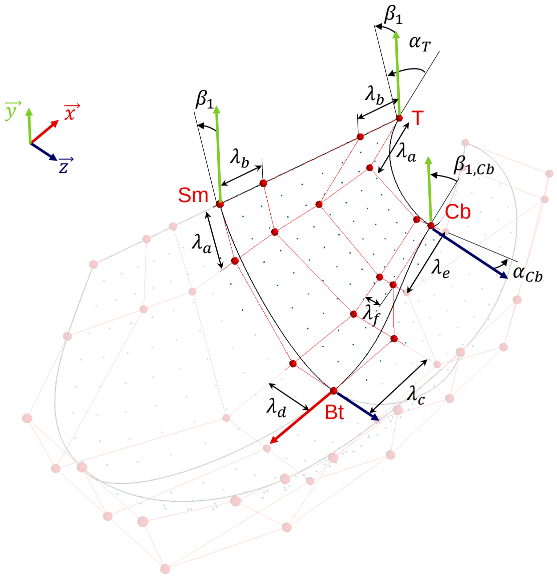


Figure A.4 – Control points defining the first Bézier surface [4].

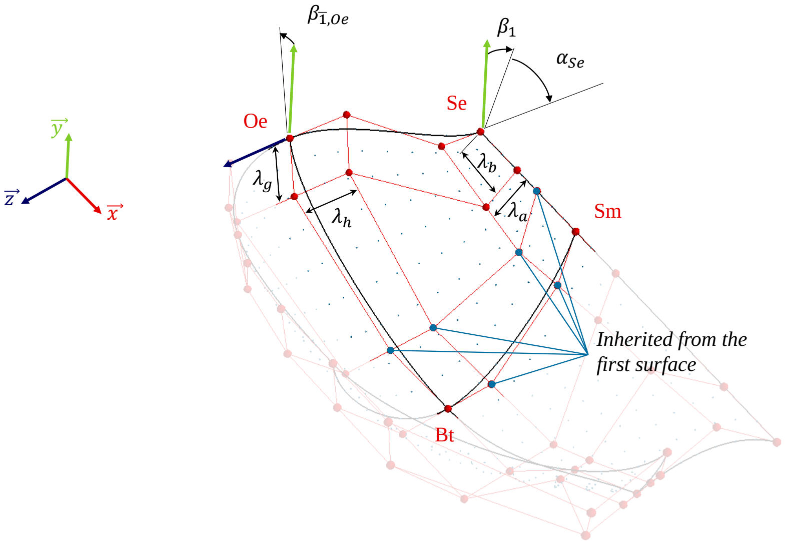


Figure A.5 – Control points defining the second Bézier surface [4].

Third surface

The C^1 continuity between the second and the third surface directly defines 8 of the 16 control points of the second surfaces, as shown in Figure A.6. Only three vertex of the third surface are physical points. The fourth vertex named Or and the associated edge orientation are built according to the definition given in Figure A.7 with three scalar parameters ϕ_{Or}, ψ_{Or} and χ_{Or}. The outlet surface is oriented with the angles $\beta_{\bar{1},Om}$ and $\beta_{\bar{1},Or}$.

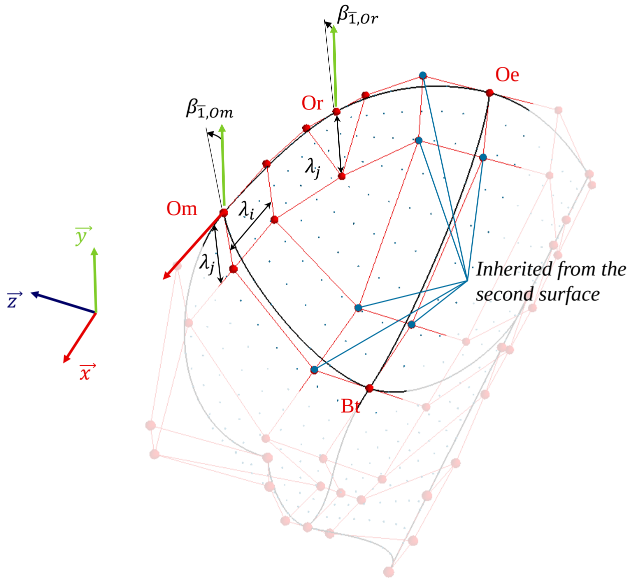


Figure A.6 – Control points defining the third Bézier surface [4].

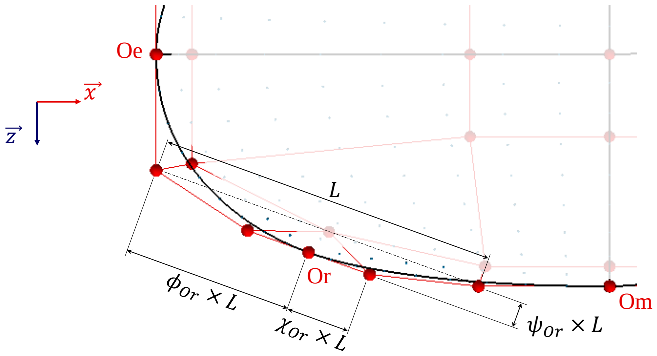


Figure A.7 – Definition of the fourth vertex of the third Bézier surface with respect to physical points [4].

Fourth surface

For the fourth surface, only the three control points around Ce are still free. The other control points are inherited from the other surfaces to ensure continuity. The outlet edge is oriented with an angle α_{Ce} around Y in Ce and the cutout edge in Ce is oriented directly along Y. The position of the other control points are defined by the two distances $L_{u,Ce}$ and $L_{v,Ce}$ represented in Figure A.8.

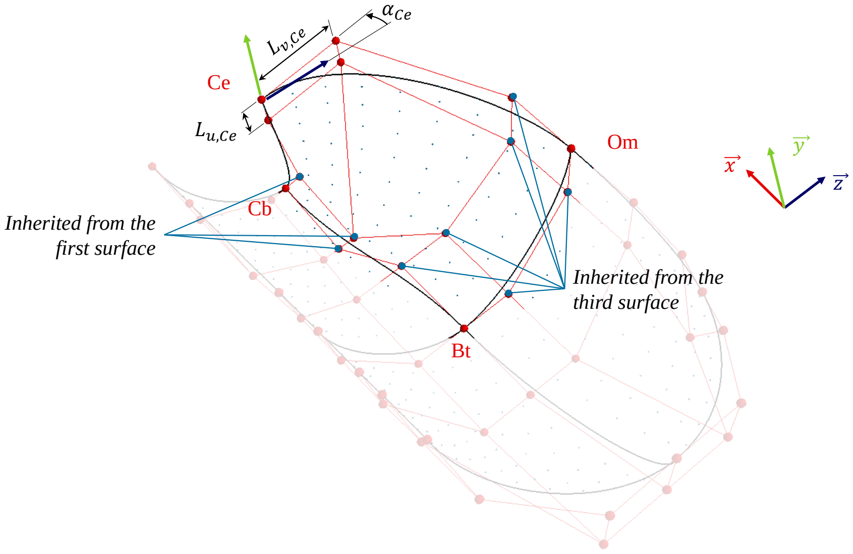


Figure A.8 – Control points defining the fourth Bézier surface [4].

Definition of the outer surface

The outer surface is built by offsetting the inner surface along its local normals with a given thickness. P_{uv} represents a point of the inner surface with the parameters (u, v). The normal at this point is written $\boldsymbol{n}_{uv}$. Given a thickness function $\mathscr{T}$ defined on $[0,1]^2$ that returns a thickness for each (u, v) parameters, then the point P'_{uv} of the outer surface associated to P_{uv}, is given by:

$$P'_{uv} = P_{uv} + \mathscr{T}(u, v) \cdot \boldsymbol{n}_{uv} \tag{A.3}$$

For each surface of the bucket inner wall, a thickness map is defined by a set of $m \times n$ *control thickness* t_{ij}. The thickness function is of the form defined in (A.5).

$$\mathscr{T} : \left| \begin{array}{ccl} [0,1]^2 & \longrightarrow & \mathbb{R} \\ u, v & \longmapsto & \sum_{i=0}^{m} \sum_{j=0}^{n} B_{im}(u) \cdot B_{jn}(v) \cdot t_{ij} \end{array} \right. \tag{A.4}$$

Three values t_1, t_2 and t_3 are used to define the thickness maps associated to the four inner surfaces. The sets of control thickness are graphically represented in Figure A.9.

When generating the entire bucket by symmetry, the sampled points with negative coordinates along Z will be removed.

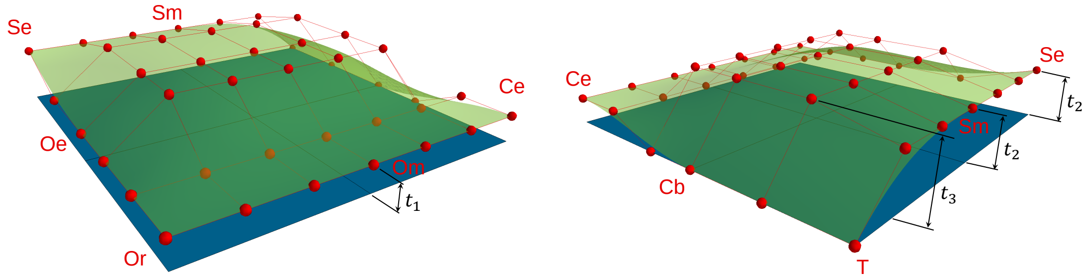


Figure A.9 – Variable thickness map defining the outer surface as an offset of the inner surface [4].

Closing the bucket volume

The inner surfaces and outer surfaces edges are collections of isoparametric curves. A pair of curve $\mathscr{C}_i$ and $\mathscr{C}_o$ with parameter written p defined on adjacent edges are considered to define the joining surface $\mathscr{S}$ between them as follows:

$$\mathscr{S} : \left| \begin{array}{ccc} [0,1]^2 & \longrightarrow & \mathbb{R} \\ p, \alpha & \longmapsto & \alpha \cdot \mathscr{C}_i(p) + (1-\alpha) \cdot \mathscr{C}_o(p) \end{array} \right. \tag{A.5}$$

This yield half of a bucket as pictured in Figure A.10.

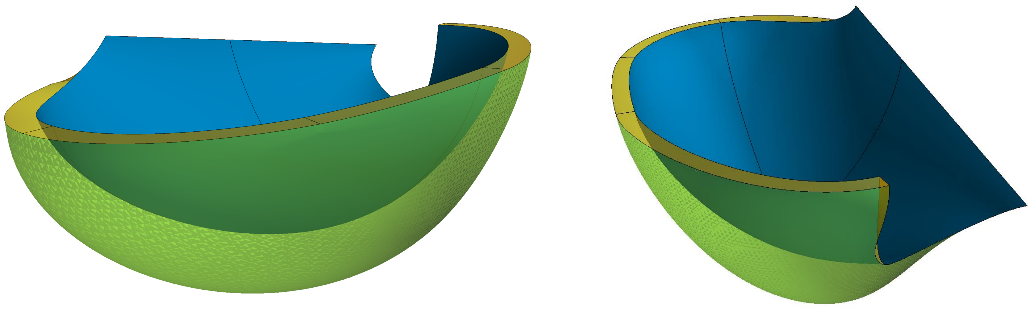


Figure A.10 – Representation of half a bucket [4].

Conclusion on the implementation of the model

The proposed model defines a Pelton bucket as a collection of parametric surfaces $\mathscr{S}(u, v)$. Given a sampling strategy for the parameters u and v for each surface adapted to the simulation requirement, a geometrical model of the bucket can be generated for CFD simulations.

This geometrical model will depend on numerous parameters inventoried in Table A.1. They are classified in four categories:

- the pre-determined parameters are the general dimensions assumed to be computed according to general charateristics of the machine;
- the optimisation parameters that constitute the search space for the Pelton bucket optimisation;
- the fixed parameters, that are arbitrarily fixed and that are assumed to have a minor impact on the Pelton bucket performances;
- the internally optimised parameters defined in the following paragraph.

Table A.1 – Inventory of the parameters describing the Pelton bucket geometry.

Category	**Parameters**
Pre-determined	A_1, A_2, B, C_2, C_3, E and F
Optimisation	y_T, α_T, ι, β_1, $\beta_{1,Cb}$, x_{Cb}, y_{Cb}, z_{Cb}, α_{Ce}, $L_{u,Ce}$ and $L_{v,Ce}$
Fixed	$\lambda_\bullet$
Internally optimised	$\beta_{\bar{1},Oe}$, $\beta_{\bar{1},Om}$ and $\beta_{\bar{1},Or}$

Internally optimized parameters

The Pelton bucket performance are known to increase with a decrease of the outlet angles $\beta_{\bar{1}}$. The Figure A.11 illustrates the limit of this assertion: up to a certain angle, the water jet impinges on the next bucket with an associated loss of energy.

With all the other parameters fixed, and considering the output water jet of known thickness tangent to the outlet surface, the optimal $\beta_{\bar{1}}$ can then be searched to provide the minimal values without collision.

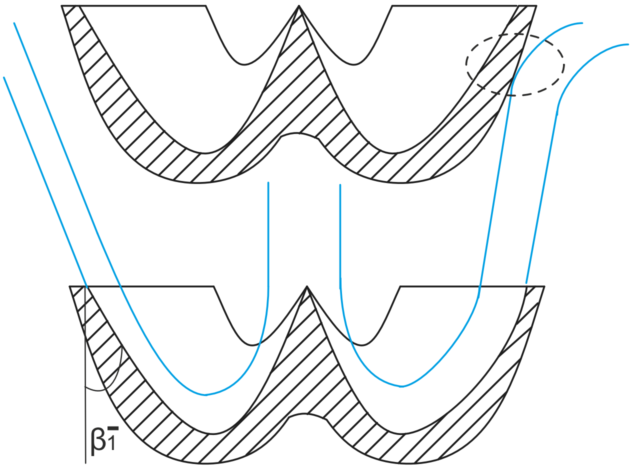


Figure A.11 – Illustration of the heeling phenomenon [4].

Bibliography

[1] S. Adami, X. Hu, and N. Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012.

[2] S. Adami, X. Hu, and N. Adams. A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 241:292–307, 2013.

[3] J. S. Anagnostopoulos and D. E. Papantonis. A fast Lagrangian simulation method for flow analysis and runner design in Pelton turbines. *Journal of Hydrodynamics*, 24:930–941, 2012.

[4] L. Andolfatto. Pelton parametric model. Technical report, EPFL, Laboratory for Hydraulic Machines, 2014.

[5] M. Antuono, A. Colagrossi, S. Marrone, and D. Molteni. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Computer Physics Communications*, 181:532–549, 2010.

[6] H. Atkins. *Continued development of the discontinuous Galerkin method for computational aeroacoustic applications.* American Institute of Aeronautics and Astronautics, 1997.

[7] S. Atluri and T. Zhu. A new meshless local petrov-galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22(2):117–127, 1998.

[8] F. Avellan. Hydraulic turbomachines formulary. Technical report, EPFL, Laboratory for Hydraulic Machines, 2010.

[9] F. Avellan, P. Dupont, S. Kvicinsky, L. Chapuis, E. Parkinson, and G. Vullioud. Flow calculations in Pelton turbines. Part 2: Free surface flows. In *Proceedings of the 19th IAHR Symposium on Hydraulic Machinery and Cavitation, Singapore*, pages 294–305, 1998.

[10] M. Basa, L. Lobovsky, and N. J. Quinlan. Parallelisation of a finite volume particle method code. In *Proceedings of the 7th International SPHERIC Workshop, Prato*, pages 282–286, 2012.

[11] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994.

[12] F. Bierbrauer, P. Bollada, and T. Phillips. A consistent reflected image particle approach to the treatment of boundary conditions in smoothed particle hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 198(41):3400–3410, 2009.

[13] G. C. Chaudhari, S. P. Shah, D. B. Kulshreshatha, and S. Channiwala. Pressure distribution inside the fixed bucket of Pelton wheel. *International Journal of Mechanical Engineering and Research*, 3:41–48, 2013.

[14] J. Chen, J. Beraun, and T. Carney. A corrective smoothed particle method for boundary value problems in heat conduction. *International Journal for Numerical Methods in Engineering*, 46(2):231–252, 1999.

[15] M. Cheng and G. Liu. A novel finite point method for flow simulation. *International Journal for Numerical Methods in Fluids*, 39(12):1161–1178, 2002.

[16] J. M. Cherfils, G. Pinon, and E. Rivoalen. JOSEPHINE: A parallel SPH code for free-surface flows. *Computer Physics Communications*, 183:1468–1480, 2012.

[17] G. A. Dilts. Moving least-squares-particle hydrodynamics i: consistency and stability. *International Journal for Numerical Methods in Engineering*, 44(8):1115–1155, 1999.

[18] G. A. Dilts. Moving least-squares-particle hydrodynamics ii: conservation and boundaries. *International Journal for Numerical Methods in Engineering*, 48(10):1503–1524, 2000.

[19] J. Fang, A. Parriaux, M. Rentschler, and C. Ancey. Improved SPH methods for simulating free surface flows of viscous fluids. *Applied Numerical Mathematics*, 59(2):251–271, 2009.

[20] R. Fatehi and M. Manzari. A consistent and fast weakly compressible smoothed particle hydrodynamics with a new wall boundary condition. *International Journal for Numerical Methods in Fluids*, 68(7):905–921, 2012.

[21] 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.

[22] M. Gomez-Gesteira, B. D. Rogers, R. A. Dalrymple, A. J. C. Crespo, and M. Narayanaswamy. *User guide for the SPHysics code v2.0*, 2010.

[23] D. Guibert, M. de Leffe, G. Oger, and J. G. Piccinali. Efficient parallelization of 3D SPH schemes. In *Proceedings of the 7th International SPHERIC Workshop, Prato*, pages 259–265, 2012.

[24] M. Guilbaud, J. B. Houdeline, and R. Philibert. Study of the flow in the various sections of a Pelton turbine. In *Proceedings of the 16th IAHR Symposium on Hydraulic Machinery and Cavitation, São Paulo*, pages 819–831, 1992.

[25] H. Hana. A discussion on numerical simulation in Pelton turbines. In *Proceedings of the 19th IAHR Symposium on Hydraulic Machinery and Cavitation, Singapore*, pages 306–314, 1998.

[26] D. Hietel, K. Steiner, and J. Struckmeier. A finite-volume particle method for compressible flows. *Mathematical Models and Methods in Applied Sciences*, 10:1363–1382, 2000.

[27] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[28] D. W. Holmes, J. R. Williams, and P. Tilke. A framework for parallel computational physics algorithms on multi-core: SPH in parallel. *Advances in Engineering Software*, 42:999–1008, 2011.

[29] T. Ishiyama, T. Fukushige, and J. Makino. GreeM: Massively parallel TreePM code for large cosmological N-body simulations. *Publications of the Astronomical Society of Japan*, 61:1319–1330, 2009.

[30] E. Jahanbakhsh. *Simulation of silt erosion using particle-based methods.* PhD thesis, EPFL n° 6284, 2014.

[31] E. Jahanbakhsh, O. Pacot, and F. Avellan. Improving accuracy of viscous fluid simulation using finite particle method. In *Proceedings of the 6th International SPHERIC Workshop, Hamburg*, pages 16–138, 2011.

[32] E. Jahanbakhsh, O. Pacot, and F. Avellan. Implementation of a parallel SPH-FPM solver for fluid flows. *Zetta, numerical simulation for science and technology*, 1:16–20, 2012.

[33] E. Jahanbakhsh, C. Vessaz, and F. Avellan. Finite volume particle method for 3-D elasto-plastic solid simulation. In *Proceedings of the 9th International SPHERIC Workshop, Paris*, pages 356–362, 2014.

[34] E. Jahanbakhsh, C. Vessaz, and F. Avellan. Silt motion simulation using finite volume particle method. In *Proceedings of the 27th IAHR Symposium on Hydraulic Machinery and Systems, Montréal*, 2014.

[35] B. Janetzky, E. Gode, A. Ruprecht, H. Keck, and C. H. Scharer. Numerical simulation of the flow in a Pelton bucket. In *Proceedings of the 19th IAHR Symposium on Hydraulic Machinery and Cavitation, Singapore*, pages 276–284, 1998.

[36] D. Jost, P. Meznar, and A. Lipej. Numerical prediction of a Pelton turbine efficiency. In *Proceedings of the 25th IAHR Symposium on Hydraulic Machinery and Systems, Timisoara*, 2010.

[37] P. Koumoutsakos. Multiscale flow simulations using particles. *Annual Review of Fluid Mechanics*, 37:457–487, 2005.

[38] S. Kvicinsky. *Méthode d'analyse des écoulements 3D à surface libre: Application aux turbines Pelton.* PhD thesis, EPFL n° 2526, 2002.

[39] S. Kvicinsky, J. L. Kueny, and F. Avellan. Free surface flows: experimental validation of volume of fluid (VOF) method in the plane wall case. In *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference, San Francisco, California*, 2002.

[40] S. Kvicinsky, J. L. Kueny, F. Avellan, and E. Parkinson. Experimental and numerical analysis of free surface flows in a rotating bucket. In *Proceedings of the 21st IAHR Symposium on Hydraulic Machinery and Systems, Lausanne*, 2002.

[41] S. Kvicinsky, F. Longatte, J. Kueny, and F. Avellan. Numerical and experimental analysis of free surface flow in a 3D non rotating pelton bucket. In *Proceedings of the 9th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery, Honolulu, Hawaii*, 1999.

[42] M. S. Liou. A sequel to AUSM: AUSM+. *Journal of Computational Physics*, 129:364–382, 1996.

[43] T. Liszka, C. Duarte, and W. Tworzydlo. hp-meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 139:263–288, 1996.

[44] G.-R. Liu and M. B. Liu. *Smoothed particle hydrodynamics: a meshfree particle method.* World Scientific, 2003.

[45] M. Liu and G. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering*, 17(1):25–76, 2010.

[46] M. Liu, W. Xie, and G. Liu. Modeling incompressible flows using a finite particle method. *Applied Mathematical Modelling*, 29(12):1252–1270, 2005.

[47] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20:1081–1106, 1995.

[48] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

[49] R. Mack and W. Moser. Numerical investigation of the flow in a Pelton turbine. In *Proceedings of the 21st IAHR Symposium on Hydraulic Machinery and Systems, Lausanne*, pages 373–378, 2002.

[50] J. C. Marongiu, F. Leboeuf, J. Caro, and E. Parkinson. Low Mach number numerical schemes for the SPH-ALE method. Application in free surface flows in Pelton turbines. In *Proceedings of the 4th International SPHERIC Workshop, Nantes*, pages 323–330, 2009.

[51] J. C. Marongiu, F. Leboeuf, J. Caro, and E. Parkinson. Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method. *Journal of Hydraulic Research*, 48:40–49, 2010.

[52] J. C. Marongiu, F. Leboeuf, 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. In *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, volume 221, pages 849–856. SAGE Publications, 2007.

[53] J. C. Marongiu, F. Leboeuf, and E. Parkinson. Riemann solvers and efficient boundary treatments: an hybrid SPH-finite volume numerical method. In *Proceedings of the 3rd International SPHERIC Workshop, Lausanne*, pages 101–108, 2008.

[54] J. C. Marongiu, E. Parkinson, S. Lais, F. Leboeuf, and J. Leduc. Application of SPH-ALE method to Pelton hydraulic turbines. In *Proceedings of the 5th International SPHERIC Workshop, Manchester*, pages 253–258, 2010.

[55] S. Marrone, A. Colagrossi, M. Antuono, G. Colicchio, and G. Graziani. An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers. *Journal of Computational Physics*, 245:456–475, 2013.

[56] D. Molteni and A. Colagrossi. A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH. *Computer Physics Communications*, 180(6):861–872, 2009.

[57] J. Monaghan and J. Kajtar. SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(10):1811–1820, 2009.

[58] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.

[59] J. J. Monaghan. SPH without a tensile instability. *Journal of Computational Physics*, 159(2):290–311, 2000.

[60] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, 2005.

[61] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1):214–226, 1997.

[62] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5):307–318, 1992.

[63] R. M. Nestor, M. Basa, M. Lastiwka, and N. J. Quinlan. Extension of the finite volume particle method to viscous flow. *Journal of Computational Physics*, 228(5):1733–1749, 2009.

[64] E. Parkinson, R. Lestriez, and L. Chapuis. Flow calculations in Pelton turbines - Part 1 Repartitor and injector numerical analysis. In *Proceedings of the 19th IAHR Symposium on Hydraulic Machinery and Cavitation, Singapore*, pages 285–293, 1998.

[65] A. Perrig. *Hydrodynamics of the free surface flow in Pelton turbine buckets.* PhD thesis, EPFL n° 3715, 2007.

[66] A. Perrig, F. Avellan, J.-L. Kueny, M. Farhat, and E. Parkinson. Flow in a Pelton turbine bucket: numerical and experimental investigations. *Journal of Fluids Engineering*, 128(2):350–358, 2006.

[67] A. Perrig, M. Farhat, and F. Avellan. High speed flow visualisation of an impinging jet on a Pelton turbine bucket. In *Proceedings of the 5th Joint ASME/JSME Fluids Engineering Conference, San Diego, California*, 2004.

[68] A. Perrig, M. Farhat, F. Avellan, E. Parkinson, H. Garcin, C. Bissel, M. Valle, and J. Favre. Numerical flow analysis in a Pelton turbine bucket. In *Proceedings of the 22nd IAHR Symposium on Hydraulic Machinery and Systems, Stockholm*, 2004.

[69] A. Perrig, M. Valle, M. Farhat, E. Parkinson, J. Favre, and F. Avellan. Onboard flow visualization in a pelton turbine bucket. In *Proceedings of the 23rd IAHR Symposium on Hydraulic Machinery and Systems, Yokohama*, 2004.

[70] N. J. Quinlan and R. M. Nestor. Fast exact evaluation of particle interaction vectors in the finite volume particle method. *Meshfree Methods for Partial Differential Equations V*, pages 219–234, 2011.

[71] M. Reclari. *Hydrodynamics of orbital shaken bioreactors.* PhD thesis, EPFL n° 5759, 2013.

[72] A. Santolin, G. Cavazzini, G. Ardizzon, and G. Pavesi. Numerical investigation of the interaction between jet and bucket in a Pelton turbine. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 223:721–728, 2009.

[73] V. Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, pages 1105–1134, 2005.

[74] D. Valdez-Balderas, J. M. Dominguez, A. J. C. Crespo, and B. D. Rogers. Developing massively parallel SPH simulations on multi-GPU clusters. In *Proceedings of the 6th International SPHERIC Workshop, Hamburg*, pages 340–347, 2011.

[75] C. Vessaz, E. Jahanbakhsh, and F. Avellan. FPM simulations of a 3D impinging jet on a flat plate comparison with CFD and experimental results. In *Proceedings of the 7th International SPHERIC Workshop, Prato*, pages 214–220, 2012.

[76] C. Vessaz, E. Jahanbakhsh, and F. Avellan. FPM flow simulations using an adaptive domain decomposition strategy. In *Proceedings of the 8th International SPHERIC Workshop, Trondheim*, pages 227–232, 2013.

[77] C. Vessaz, E. Jahanbakhsh, and F. Avellan. Flow simulations of a pelton bucket using finite volume particle method. In *Proceedings of the 27th IAHR Symposium on Hydraulic Machinery and Systems, Montréal*, 2014.

[78] C. Vessaz, E. Jahanbakhsh, and F. Avellan. FPM simulations of a high speed water jet validation with CFD and experimental results. In *Advances in Hydroinformatics: SIMHYDRO 2012? New Frontiers of Simulation*, pages 419–431. Springer, 2014.

[79] J. Vila. On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences*, 9(02):161–209, 1999.

[80] J. Wang and G. Liu. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54(11):1623–1648, 2002.

[81] Y. X. Xiao, T. Cui, Z. W. Wang, and Z. G. Yan. Numerical simulation of unsteady free surface flow and dynamic performance for a Pelton turbine. In *Proceedings of the 26th IAHR Symposium on Hydraulic Machinery and Systems, Beijing*, 2012.

[82] Y. X. Xiao, F. Q. Han, J. L. Zhou, and T. Kubota. Numerical prediction of dynamic performance of Pelton turbine. *Journal of Hydrodynamics*, 19:356–364, 2007.

[83] R. Xu, P. Stansby, and D. Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics*, 228(18):6703–6725, 2009.

[84] X. Zhang, X.-H. Liu, K.-Z. Song, and M.-W. Lu. Least-squares collocation meshless method. *International Journal for Numerical Methods in Engineering*, 51(9):1089–1100, 2001.

[85] Y. Zhu and R. Bridson. Animating sand as a fluid. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 965–972, 2005.

[86] B. Zoppe, C. Pellone, T. Maitre, and P. Leroy. Flow analysis inside a Pelton turbine bucket. *Journal of Turbomachinery*, 128:500–511, 2006.

Curriculum Vitae

Christian Vessaz

Rue du Midi 5	24.07.1986
1040 Villars-le-Terroir	swiss
Switzerland	married
christian.vessaz@a3.epfl.ch	+41 79 741 40 75

Education

2011 - 2014	**Doctoral Thesis, PhD** in energy EPFL, École polytechnique fédérale de Lausanne, Switzerland
2008 - 2010	**Master of Science, MSc** in mechanical engineering EPFL, École polytechnique fédérale de Lausanne, Switzerland
2005 - 2008	**Bachelor of Science, BSc** in mechanical engineering EPFL, École polytechnique fédérale de Lausanne, Switzerland
2002 - 2005	**Maturité fédérale** Gymnase d'Yverdon-les-Bains, Switzerland Awards: Mathématiques, Physique et application des mathématiques

Professional experience

2010 - 2014	**Doctoral assistant** - EPFL, Laboratory for Hydraulic Machines Numerical simulations of hydraulic machines Participation in teaching activities as an assistant Supervision of semester projects and master theses
2010 (6 months)	**Master Thesis** - Voith Hydro, Heidenheim an der Brenz, Germany Transient CFD Simulation of a Francis Turbine using OpenFOAM
2007 - 2008 (20%)	**Internship** - RM Synergie SA, Corcelles-près-Payerne, Switzerland Realization of different mechanical parts from blueprints Measurement of mechanical parts and settings of CNC turning machines

Computing skills

Programming:	C, C^{++}, Python, Shell, Matlab, Git, SLURM, PBS pro, LoadLeveler
CFD / CAD:	Ansys CFX, ParaView, OpenFOAM, Catia
Office:	MS-Office, LaTeX
Multimedia:	Adobe Illustrator

Languages

French:	Native speaker
English:	Advanced (C1)
German:	Intermediate (B1)

Hobbies and interest

Running, RC model aircraft and member of local societies

Journal papers

C. Vessaz, E. Jahanbakhsh and F. Avellan. FVPM Flow Simulation of Jet Deviation by Rotating Pelton Buckets, *submitted in Journal of Fluids Engineering*, September, 2014

C. Vessaz, E. Jahanbakhsh and F. Avellan, Finite particle simulation of a high-speed water jet impinging on a flat plate, *submitted in Journal of Computational Physics*, August, 2013

C. Vessaz, C. Tournier, C. Münch and F. Avellan, Design optimization of a 2D blade by means of milling tool path, *CIRP Journal of Manufacturing Science and Technology*, volume 6, issue 3, pages 157-166, 2013

Conference papers

C. Vessaz, E. Jahanbakhsh and F. Avellan, Flow simulation of a Pelton bucket using finite volume particle method, *7th IAHR Symposium on Hydraulic Machinery and Systems*, Montréal, Canada, September 22-26, 2014
Award: Student best paper award by ALSTOM

E. Jahanbakhsh, C. Vessaz and F. Avellan, Silt motion simulation using Finite Volume Particle Method, *7th IAHR Symposium on Hydraulic Machinery and Systems*, Montréal, Canada, September 22-26, 2014

E. Jahanbakhsh, C. Vessaz and F. Avellan, Finite Volume Particle Method for 3-D Elasto-Plastic Solid Simulation, *9th international SPHERIC workshop*, CNAM, Paris, France, June 3-5, 2014

C. Vessaz, E. Jahanbakhsh and F. Avellan, FPM Simulations of a High Speed Water Jet Validation with CFD and Experimental Results, *Advances in Hydroinformatics: SIMHYDRO 2012 – New Frontiers of Simulation*, pages 419-431, Springer, 2014

E. Jahanbakhsh, C. Vessaz and F. Avellan, Simulation of Impinging Jet using Finite Volume Particle Method, *5th International IAHR Workshop on Cavitation and Dynamic Problems in Hydraulic Machinery*, EPFL, Lausanne, Switzerland, September 8-11, 2013

C. Vessaz, E. Jahanbakhsh and F. Avellan, FPM Flow Simulations Using an Adaptive Domain Decomposition Strategy, *8th international SPHERIC workshop*, SINTEF, Trondheim, Norway, June 4-6, 2013

C. Vessaz, E. Jahanbakhsh and F. Avellan, FPM Simulations of a 3D Impinging Jet on a Flat Plate-Comparison with CFD and Experimental Results, *7th international SPHERIC workshop*, Monash University, Prato, Italy, May 29-31, 2012