Implementation of the Moving Particle Semi-implicit method to predict the drag resistance coefficient on 2D

PhD thesis

by

Carlos Andrés Pérez Gutiérrez

Supervisor

Manuel Julio García Ruiz, Ph.D.

Facultad de Ingeniería
Universidad Eafit
Medellín, June 2016

To my mother who always believed in me

To my family who supported me

To all those people whom I have learnt from

'What do you know about this business?' the King said to Alice.

'Nothing,' said Alice.

'Nothing ***whatever****?' persisted the King.*

'Nothing whatever,' said Alice.

'That's very important,' the King said, turning to the jury.

*...the White Rabbit interrupted: '****Un****important, your Majesty means, of course,'...*

. ... 'but it doesn't matter a bit,' she thought to herself.

Alice's Adventures in Wonderland, Lewis Carroll (1832-1898)

Acknowledgements

I would like to thank to my supervisor Manuel Julio García for his academic advise, personal support and specialy for his kindness, without him this would have been impossible. Also, all the members of the Applied Mechanics Research Group, the Apolo computing centre and the Hydraulic laboratory at Eafit University.

Abstract

A dam break problem and the flow around a 2D submerged body on different scenarios were solved with the original Moving Particle Semi-implicit (MPS) method proposed by Koshizuka and Oka in 1996. The results of this study showed that although the original method reproduces the free surface of the fluid on the dam break computation, it can not accurately compute the pressure distribution over the submerged bodies. It was found that the free surface was inaccurate when negative pressures were present in the particle domain. Also, when modelling the interaction of a solid immersed in a fluid, the simulation exhibited stability issues and solid penetration. Several modifications of the original MPS were studied, implemented and tested. This thesis proposes a modified Moving Particle Semi-implicit (MPS) method for modelling immerse bodies in an free surface flow. The MPS method is based on the prediction-correction calculation of the velocity field based on the Helmhotz-Hodge decomposition. Initially the predicted velocity is calculated based on the viscous and external forces terms and then corrected by the gradient of the pressure which is obtained by the solution of the Poisson Pressure's equation. This thesis shows how small variations in the source term of the Poisson Pressure's equation can destabilise or stabilise simulations. One of the main result of this research is an improved stability by means of a reformulation of the Poisson Pressure equation and the aid of relaxation factors. Also, the pressure gradient was computed for non free surface particles only. The results show that, although pressure fluctuations were still present, good results were obtained when compared the drag coefficient to the reported values in the literature.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Definition, description of the problem and aim

In fluid dynamics, the drag resistance is the force acting on the opposite direction of an object when it is moving into a fluid or when there is a flow against a body. Hence, resistance calculations are strongly linked with the power required to move objects through such a fluid flow and also with forces necessaries to keep static structures standing. Over the years, a large variety of studies have been carried out with bodies completely immersed in the fluid. Nonetheless, within the hydrodynamics field, bodies have to move or stand with the presence of a free surface. Although, current numerical mesh methods calculate the drag resistance the determination of the free surface is a quite complex process which requires complex mesh algorithms. This thesis aims to implement a numerical method which allows to predict the drag resistance over a 2D object immersed in a fluid flow and to visualise accurately the free surface.

1.2 Motivation and difficulties

In former studies of computational resistance with free surface carried out in a commercial code, based on a Finite Volume method with Reynolds Average Navier Stokes' Equations, the following disadvantages were found (Pérez-Gutiérrez[24]): A grid is necessary to calculate spatial derivatives and it is important to numerical accuracy and stability, mesh independence studies are necessary and also time consuming, large deformations causes inadequate mesh qualities, the free surface had to be known beforehand in order to refine the mesh on the corresponding part of the domain, on the post processing stage the free surface was not clearly defined as it changed depending on the volume fraction of air and water defined by the user and according to Versteg and Malalasekera (1995) numerical diffusion is prone due to the first order accuracy discretisation. Numerical diffusion was also reported by Azcueta [2] on the wave amplitude reduction of a free surface wave traveling through a domain using Reynolds Averaged Navier Stokes Equation. On the contrary, the advantages of the above mentioned method are related with the optimisation through years which allows to obtain trustworthy results quickly.

Within the deterministic mesh-less methods it was found on the literature the Smoothed Particle Hydrodynamics SPH and the Moving Particle Semi-Implicit Method MPS. The first is an explicit method and the second a semi-implicit one which was thought as an advantage as it allowed large time step sizes. Other advantage of the MPS is that it was designed for incompressible fluids, meanwhile SPH was initially conceived for compressible fluids, to adjust the model for incompressible fluids it uses an equation of state with a non realistic speed of sound associated to a very small time step [4] and also artificial viscosity. Thus, after several attempts to effectively compute the drag resistance and the free surface behaviour with the finite volume method, the Moving Particle Semi Implicit (MPS) method proposed by Koshizuka and Oka [15] was implemented and tested.

According to Koshizuka and Oka [15] the advantages of the MPS are: It does not need explicit surface tracking, grid is not necessary, numerical diffusion does not occur because it is a fully Lagrangian scheme, hence the advection term is not taking into account and the convection is calculated by the motion of particles. Simulations have the capability to analyse more complex geometry and physics than grid methods with either the finite volume or the finite element method. It is easy to add or remove particles in the middle of a simulation. On the other hand, the main disadvantage of the original MPS method is the unphysical pressure fluctuations. The method is not robust enough and can be destabilised on the presence of negative pressures, hence numerical instabilities showing unphysical particle behaviour ended up the simulation. Also, generation of neighbour lists are quite time consuming [14]. According to our studies the neighbour searching is the highest percentage of time iterations, between 35% and 37% of the total time step and the fact that the fluid incompressibility is not always satisfied; i.e. particle number density calculated at the end of each time step for every single particle is not equal to the initial one.

The MPS method has been actively developed in the last years. For instance, Khayyer and Gotoh [7, 8] proposed what they had named Corrected MPS, which consist of forcing the gradient to obey the linear momentum conservation by making forces between particles symmetric, this is, equal in magnitude and opposite in direction. Nonetheless, they have stated in 2011 [10] that non-exact momentum conservation give less numerical errors than those corresponding to approximation of the pressure gradient, and turn back back to the original MPS gradient calculation with an additional Gradient Correction (GC). Further in 2013, Tsuruta, Khayyer and Gotoh [12] state clearly that previous corrections on the gradient implemented by themselves in 2011, referring to GC, do not resolve the maldistribution of particles and it requires a careful setting of calculation conditions. Thus, they proposed a Dynamically Stabilised (DS) gradient operator as a repulsive force to stabilise the calculation.

Moreover, in 2014, Hwang with Khayer, Gotoh and Park [5], did not use the DS gradient operator, they used the GC matrix for the Poisson Pressure Equation (PPE) and also adopted the source term proposed by Tanaka and Masunaga [31]. This source term has two components: the divergence of the intermediate velocity plus the fraction of the particle number density deviation with a relaxation factor needed in order to run stable simulations. However, this source term had already been criticised by Khayyer and Gotoh [10] by containing the above mentioned relaxation factor, and due to the discretization of that source term simulations like the dam break still revealed considerable numerical noise and non-pyshical behaviour. Hwang et al [5] has quote Lee et al [16] summarising that the particle method is not considered mature enough and fully reliable computational method due to non-physical pressure fluctuations and long computing time. The last issue is being compensated nowadays with parallel computing.

In the same way, Khayyer and Gotoh [10] criticised the use of unknown coefficients in the three components of the source term of the PPE proposed by Kondo and Koshizuka [13] as those coefficients were obtained by hydrostatic pressure calculations. Thus, Khayyer and Gotoh [10, 12] used a three terms source for the PPE and called the Error Compensation Source term. The first term is a high source term proposed in 2009 [8] and the other two terms are dynamic instantaneous flow coefficients. However, Tamai and Koshizuka [30] reject their source term arguing that its accuracy is less than zero "since they adopted non-renormalized SPH divergence operator to approximate time derivative of particle number density." Hence, the way to determine which implementation works well has been tough as there must be something hidden behind simulations where authors are not clear enough, the MPS method is under construction and it can work for a specific situation by a carefully setting up of conditions.

1.3 Objectives

1.3.1 Main Objective

To predict the drag coefficient of an immersed 2D body close to a free surface by means of the implementation of a stable Moving Particle Semi-Implicit Method.

1.3.2 Specific Objectives

- To implement the Moving Particle semi-implicit method on Python 3.2.
- To evaluate all the variants of the MPS found in literature on the flow around a 2D body.
- To develop a new stable form of the MPS in order to correctly predict the behaviour of the free surface and to properly compute the drag forces.

1.4 Methodology

The original MPS method was implemented in Python 3.2 in order to test its capabilities. The dam break benchmark was used to compare results.The following modifications of the MPS were implemented and studied in order to evaluate its accuracy and stability.

- The correction for the gradient or CMPS [7] where the linear momentum between particles is theoretically conserved.
- The High Source term (HS) on the Poisson Pressure's equation (PPE) [8].
- The High order Laplacian (HL) and High order Viscous term (HV) [9].
- The two terms for the PPE [31].
- Error Compensation for the source term(ECS) on the PPE and the gradient correction (GC) matrix [10].

The following aspects were evaluated, based on the main issues of the MPS, as several authors agree, such as:

- Particle behaviour according to the laws of physics .
- Maldistribution of particles meanwhile simulations advance in time.
- Pressure oscillations.

- Negative pressures
- Algorithm stability.

A modified MPS was proposed and tested in order deal with those issues as well to predict the numerical drag coefficient resistance. A 2D square geometry was chosen as it was found experimental data [33] with free surface showing similar values as the full submerged body. Qualitative and not quantitative experimental visualisation was carried out as the last one is out of the scope of this research and the current state of the hydraulic laboratory does not allow precision on data acquisition.

Chapter 2

Mesh-free particle methods

The particle methods can be classified either in probabilistic models such as lattice Boltzmann and molecular dynamics or deterministic models such as Smoothed Particle Hydrodynamics and Moving Particle Semi-implicit method [15]. The first methods represent macroscopic properties as statistical behaviour of microscopic particles, the second ones represent the fluid as a group of particle interactions. The main idea of a deterministic method is to replace the fluid by a set of mathematical points named particles which follow the motion and carry fluid properties with them such as mass, momentum and pressure. The fluid dynamic equations are numerically solved for those particles, or in mathematical words, those interpolation points. In this chapter a brief introduction to the SPH is given as a way to understand the particle methods, the original MPS and its variants are presented in chronological order as they were found.

2.1 Smoothed Particle Hydrodynamics, SPH

One of the first Lagrangian mesh free particle methods was the Smoothed Particle Hydrodynamics, SPH, proposed by Gingold and Monaghan in 1977 [3] to determine the fluid dynamics in astrophysics. The term smooth refers to the use of a continuous weight function derivative to prevent large fluctuation in the calculated force (Li and Liu [17]). Similarly, by the same time, Lucy [18] proposed and experimented with an analogous method for gas dynamic problems in astronomy. The original idea was thought for a compressible inviscid fluid and hence the reason from the derivation of the pressure from the equation of state. Monagham [21] showed its extension to simulation of free surface incompressible flows, presenting examples for the evolution of a drop, the dam break and the propagation of waves towards a beach.

The heart of the SPH method is the interpolation [20] which is used to determine the influence of properties of adjacent particles over the particle of interest. It uses the integral representation of a scalar function. Lets $A = A(x)$ an arbitrary function, then its integral representation is given by

$$A(x) = \int_{\Omega} A(y)\delta(x-y)dy$$

where δ is the delta Dirac function defined by

$$\delta(x-y) = \begin{cases} \infty & x = y \\ 0 & x \neq y \end{cases}.$$

SPH replace de delta Dirac function by a smooth function W, which is called weighting function or interpolating kernel,

$$A(r) = \int_{\Omega} A(y)W(r-y,h)dy \tag{2.1}$$

here the integral is over the domain Ω and dy is an element of volume in the domain. The desirable properties of W are:

1. Compactness of W, that is $W(r-y,h) - 0$ if $\frac{\|r-y\|}{\|h\|} > 0$

2. Delta, $\delta(x)$, function properties

$$\lim_{h \to 0} W(r-y,h) = \delta(r-y)$$

3. Unity condition

$$\int_{\Omega} W(r-y,h)dy = 1$$

4. Smoothness

The kernel function has continuous second derivatives and a compact support determined by h, named smoothing length, i.e. the kernel vanishes at a finite distance h and the integral over its domain Ω should be equal to one [20]. The first golden rule [20] is to assume the kernel Gaussian.

Multiplying and dividing by the density of the particles around the particle of interest and rearranging terms, the summation interpolation can be written in a discrete form as,

$$A(r) = \sum_b m_b \frac{A_b}{\rho_b} W(r-r_b,h) \tag{2.2}$$

where the summation is over the neighbours' particles within the distance h from the particle of interest, which determines the domain of influence , in other words, h is a measure of the support domain of W where it is non zero. Equation 2.2 means that the function A at a given point is the sum of contributions from surrounding particles, weighted by the distance from each particle with W. The smoothing length has direct influence on the efficiency of the computation and the accuracy of the solution. If h is too small it implies low accuracy and if h too large it implies high computational effort.

The function A in equation 2.2 can be a scalar and if W is taken to be differentiable, the gradient of A can be determined as follows,

$$\frac{\partial A}{\partial x} = \sum_b m_b \frac{A_b}{\rho_b} \frac{\partial W}{\partial x} \tag{2.3}$$

this straightforward discretisation of spatial derivatives is not necessarily the most accurate one [22].

According to Monagham [19][20][22], in the case of the pressure, to obtain higher accuracy, the interpolation function can be written using the next identity which makes the corresponding force term symmetric and conserves the linear momentum,

$$\frac{\nabla P}{\rho} = \nabla\left(\frac{P}{\rho}\right) + \frac{P}{\rho^2}\nabla\rho \tag{2.4}$$

hence,

$$\frac{\nabla P}{\rho}_a = -\sum_b m_b\left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)\nabla_a W_{ab} \tag{2.5}$$

where the gradient∇_a is taken with respect to the particle of interest a, and $W_{ab} = W(r_a - r_b, h)$

Similarly, divergence and curl are also estimated from information about particle interactions, for instance for the velocity u they can be written for a particle a as,

$$(\nabla\cdot\mathbf{u})_a = \sum_b \frac{m_b}{\rho_b}\mathbf{u}_{ba}\cdot\nabla W_{ab} \tag{2.6}$$

$$(\nabla\times\mathbf{u})_a = \sum_b \frac{m_b}{\rho_b}\mathbf{u}_{ba}\times\nabla W_{ab} \tag{2.7}$$

where $\mathbf{u}_{ba} = \mathbf{u}_b - \mathbf{u}_a$

2.1.1 Conservation of mass and momentum

The conservation of mass for a compressible fluid can be written as follows,

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \nabla\cdot\mathbf{u} = 0 \tag{2.8}$$

and in a discrete form for a particle of interest i with neighbours j as,

$$\left(\frac{D\rho}{Dt}\right)_i = -\rho_i \sum_j \frac{m_j}{\rho_j} \mathbf{u}_{ji} \cdot \nabla_i W_{ij} \tag{2.9}$$

For an ideal fluid the momentum equation is considered, hence in its Lagrangian form,

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + \mathbf{g} \tag{2.10}$$

using the equation 2.5 the momentum equation for a particle i becomes,

$$\left(\frac{D\mathbf{u}}{Dt}\right)_i = -\sum_j m_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2}\right) \nabla_i W_{ij} + \mathbf{g} \tag{2.11}$$

2.1.1.1 Viscosity

The viscosity is important to prevent instabilities in fluid flow since single particles can move in a chaotic way [4]. The artificial viscosity proposed by Monaghan [20] is introduced in the momentum equation as

$$\left(\frac{D\mathbf{u}}{Dt}\right)_i = -\sum_j m_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} + \Pi_{ij}\right) \nabla_i W_{ij} + \mathbf{g} \tag{2.12}$$

where Π_{ij} is the viscosity term:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{c}_{ij} \mu_{ij}}{\bar{\rho}_{ij}}, & u_{ij} x_{ij} < 0 \\ 0 & u_{ij} x_{ij} \geq 0 \end{cases} \tag{2.13}$$

where $\mu_{ij} = (h u_{ij} x_{ij})/(r_{ij} + \eta^2)$,$u_{ij} = u_i - u_j$, $x_{ij} = x_i - x_j$,$\bar{c}_{ij} = (c_i + c_j)/2$ is the mean speed of sound taken from the equation of state. The parameter $\eta^2 = 0.01h^2$ is included to avoid singularities and α is a parameter chosen according to each problem, making the approach empirical [4].

2.1.1.2 Equation of state

The relation between pressure and density is assumed [21] as,

$$P = B\left[\left(\frac{\rho}{\rho_o}\right)^{\gamma} - 1\right] \tag{2.14}$$

where γ is a constant between 1 and 7, $\rho_o = 1000kg/m^3$ is the reference density. The main advantage of this approach is that there is no need to resolve any Poisson Pressure's equation, which can be time consuming and where it is necessary to identify the free surface beforehand.

The speed of sound is taken as the derivative of pressure with respect to the density

$$c^2(\rho) = \frac{\partial P}{\partial \rho} = \frac{B\gamma}{\rho_o}\left(\frac{\rho}{\rho_o}\right)^{\gamma-1} \tag{2.15}$$

The Smoothed Particle Hydrodynamics or SPH method has been reviewed as one of the deterministic models. As it was initially created for fluid dynamics of compressible and inviscid fluids like the interstellar gas, some assumptions had to be done to approach to fluid dynamics of incompressible fluids. On the other hand, the Moving Particle Semi-implicit method or MPS was created for incompressible fluids and initially validated for the collapse of a water column which was one advantage of choosing the current method. The MPS is described below.

2.2 Moving Particle Semi-implicit method, MPS

The Moving Particle Semi-implicit method was proposed by Koshizuka and Oka in 1996 [15]. The method aims for modelling an incompressible fluid as a set of interacting particles to simulate free surface scenarios like fluid fragmentation. The following are the laws and functions which are used in this method.

2.2.1 Conservation of mass and momentum

The MPS method is based on mass and momentum conservation laws. The following governing equations in Lagrangian form are considered

$$\frac{D\rho}{Dt} = 0 \tag{2.16}$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + \mathbf{g} \tag{2.17}$$

where ρ is fluid density, t is time, **u** is the velocity field, P is pressure, ν is the kinematic viscosity and **g** is the gravity as external force.

2.2.2 The weight function

Interactions between particles are limited to a finite distance with the following weight function,

$$W(r) = \begin{cases} \frac{r_e}{r} - 1 & 0 \leq r \leq r_e \\ 0 & r_e \leq r \end{cases} \tag{2.18}$$

where r is the distance between two particles and r_e is the radius of interaction or cut-off radius similar to the smoothing length in SPH.

The weight function is infinite at *r=0* which is good for avoiding clustering particles (Koshizuka et al, [14]). However, according to Kondo and Koshizuka [13], this singularity is one of the causes of the MPS instabilities and they had adopted a different weight function without singularity. Ataie-Ashtiani and Farhadi [1] had studied the effect of six different weight functions, finding that the one proposed by Shao and Lo [25] improves considerably the stability. Nonetheless, the original weight function is still used in several studies as those published by Shibata K. and Koshizuka S.[26], Sueyoshi et al [29], Khayyer A. and Gotoh H. [7][8][9][10][11][12], Shibata et al [28][27], Tanaka and Masunaga [31], Tamai and Khoshizuka [30], Hwang et al[5] .

2.2.3 Particle number density, pnd.

The particle number density is the weighted average of distances from a particle i to particles j is defined as,

$$n_i = \sum_{j \neq i} W(\mid \mathbf{r_j} - \mathbf{r}_i \mid) \tag{2.19}$$

where the contribution of the particle i itself is not taken into account. In an incompressible flow model the density is constant which is equivalent to say that the particle number density be constant, in this case denoted by n_o.

2.2.4 Modelling terms

The gradient vector of a scalar quantity between two points i and j can be approximated by the product of the variation of a physical quantity ϕ divided the distance times the unit direction as follows,

$$\left(\frac{\partial \phi}{\partial r}\right)_{\mathbf{r}_j - \mathbf{r}_i} \cong \frac{\phi_j - \phi_i}{\mid \mathbf{r}_j - \mathbf{r}_i \mid^2}(\mathbf{r}_j - \mathbf{r}_i) \tag{2.20}$$

In MPS, the gradient vector is modelled using the weight function to obtain a gradient vector of the particle i as,

$$\nabla \phi_i = \frac{d}{n^o} \sum_{j \neq i} \left[\frac{\phi_j - \hat{\phi}_i}{\mid \mathbf{r}_j - \mathbf{r}_i \mid^2}(\mathbf{r}_j - \mathbf{r}_i) W(\mid \mathbf{r}_j - \mathbf{r}_i \mid) \right] \tag{2.21}$$

where d is the number of space dimension and $\hat{\phi}_i$ is taken as the minimum value of ϕ_j within the interaction ratio. This is good for numerical stability as forces between particles are always repulsive as $\phi_j - \hat{\phi}_i$ is positive [14].

The Laplacian operator is modelled using a transient diffusion equation, where part of the quantity ϕ in particle i is distributed to its neighbours particles j using the weight function given the following discretisation,

$$\nabla^2 \phi_i = \frac{2d}{\lambda n^o} \sum_{j \neq i} [(\phi_j - \phi_i) W(\mid \mathbf{r}_j - \mathbf{r}_i \mid)] \tag{2.22}$$

where λ is a parameter to ensure that the variance increase is equal to the analytical solution [15][14]

$$\lambda = \frac{\sum_{j \neq i} \mid \mathbf{r}_j - \mathbf{r}_i \mid^2 W(\mid \mathbf{r}_j - \mathbf{r}_i \mid)}{\sum_{j \neq i} W(\mid \mathbf{r}_j - \mathbf{r}_i \mid)} \tag{2.23}$$

Equation 2.22 is used to model and discretise not only the Laplacian of the velocities from the Navier stokes equation 2.17 but also the Poisson Pressure Equation (PPE). This PPE can be obtained from the Navier-Stokes equations and it is used to enforce continuity according to Ferziger and Peric (2002). The Poisson Pressure Equation with the density approximated by the particle number density is,

$$\nabla^2 P_i^{k+1} = \frac{\rho}{\triangle t^2} \frac{n_i^* - n^o}{n^o} \tag{2.24}$$

where n_i^* is the particle number density at the intermediate position r^* (equation 3.2), after the Laplacian of the velocity and gravity are computed. How to derive this equation can be seen in Appendix A. The right hand side term of equation 2.24 is modeled according to the equation 2.22 which generates a system of equations which becomes a matrix. This matrix is sparse and it was solved by means of a library in Python 3.2 for this sort of matrixes. When a particle is detected as belonging to the free surface a Dirichlet boundary condition is imposed. The PPE will have 1 in its corresponding positions of free surface particles and and zero in the source term.However, solving this equation causes instability [13].

2.2.5 Free surface condition

The original method proposed the particle number density (pnd) as criteria to determine if a particle is on a free surface. If the pnd of a particle i n_i, is less than a constant β times the initial n^o, then the particle is identified as one on the free surface.

$$n_i < \beta n^o \tag{2.25}$$

Pressure zero is given to those particles as Dirichlet boundary condition on the PPE equation. For the present study the parameter β was taken as 0.97.

2.2.6 Wall boundary condition

Solid representation is made with boundary particles based on the idea that boundaries are formed by particles which exert forces on the fluid [21]. The wall boundaries interact with fluid particles and have pressure and velocities associated. On the original MPS method, dummy particles are used alongside the wall only to calculate the particle number density. The thickness of the solid boundary depends on the number of dummy layers which is related with the interaction distance r_e. For instance, if $r_e = 2.1 \cdot l_o$, being l_o the initial particle spacing, there will be for at least two layers of dummy particles on the wall.

2.2.7 MPS Variants

The following representative chronological modifications of the original MPS have been found in the literature so far:

i) Correction of MPS, CMPS: The correction for the discretised gradient term was proposed by Khayyer and Gotoh [7] in 2008, where the linear momentum between particles is conserved. The pressure gradient calculated between a target particle i and its neighbours j is equal and opposite to that calculated between the particle j and its neighbours including the particle i. In other words, the pressure force between a particle i and j is equal in magnitude and opposite in direction. Hence, the gradient of a scalar quantity ϕ is expressed as,

$$\nabla\phi_i = \frac{d}{n^o}\sum_{j\neq i}\left[\frac{(\phi_j+\phi_i)-(\hat{\phi_j}+\hat{\phi_i})}{\mid \mathbf{r}_j-\mathbf{r}_i\mid^2}(\mathbf{r}_j-\mathbf{r}_i)W(\mid \mathbf{r}_j-\mathbf{r}_i\mid)\right] \tag{2.26}$$

ii) High source term, HS: One modification on the source term of the Poisson Pressure's equation was proposed by Khayyer and Gotoh [8] in 2009, in which the time variation of the particle number density is obtained from the derivation of equation 2.19 respect to the time in the following way and then replaced on the PPE,

$$\frac{Dn}{Dt} = \sum_{j\neq i}\frac{DW(\mid \mathbf{r}_j-\mathbf{r}_i\mid}{Dt} \tag{2.27}$$

$$\frac{DW(\mid \mathbf{r}_j-\mathbf{r}_i\mid)}{Dt} = \frac{DW_{ij}}{Dt} = \left(\frac{\partial W_{ij}}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial x_{ij}}\frac{\partial x_{ij}}{dt} + \frac{\partial W_{ij}}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial y_{ij}}\frac{\partial y_{ij}}{dt}\right) \tag{2.28}$$

$$\frac{Dn}{Dt} = \left(-\frac{r_e}{r_{ij}^2}\frac{x_{ij}}{r_{ij}}u_{ij}^* - \frac{r_e}{r_{ij}^2}\frac{y_{ij}}{r_{ij}}v_{ij}^*\right) = -\frac{r_e}{r_{ij}}(x_{ij}u_{ij}^*+y_{ij}v_{ij}^*) \tag{2.29}$$

Hence, they have named the High Source term HS and the PPE can be written as follows,

$$\nabla^2 P_i^{k+1} = -\frac{\rho}{n^o\triangle t}\sum_{j\neq i}\frac{r_e}{r_{ij}}(x_{ij}u_{ij}^*+y_{ij}v_{ij}^*) \tag{2.30}$$

where the asterisk corresponds to an intermediate position.

iii) High order Laplacian, HL, and high order viscosity, HV: The Laplacian of a scalar field can be expressed in terms of the divergence of the gradient $\nabla^2\phi_i =$

$\nabla \cdot \nabla \phi_i$. The calculation of the Laplacian proposed by Khayyer and Gotoh [9] in 2010 uses that expression and after some mathematical work arrive to the following Laplacian for two dimensions in what they have named Higher order Laplacian HL,

$$\nabla^2 \phi_i = \frac{1}{n^o} \sum_{j \neq i} (\frac{3\phi_{ij} r_e}{r_{ij}^3}) \tag{2.31}$$

If the function ϕ is the velocity, they used this term to discretise the viscous term on the Navier Stokes' equation in what they have named High Viscosity HV.

iv) Three-terms source of the PPE: The modifications on the source term of the Poisson Pressure's equation proposed by Kondo and Koshizuka [13] in 2011 are the following,

$$\frac{1}{\rho} \nabla^2 P_i^{k+1} = \frac{n_i^* - 2n_i^k + n_i^{k-1}}{n^o \triangle t^2} + \frac{\beta}{\triangle t} \frac{n_i^k - n_i^{k-1}}{n^o \triangle t} + \frac{\gamma}{\triangle t^2} \frac{n_i^k - n^o}{n^o} \tag{2.32}$$

where the first term is the central differencing of the second order particle number density with respect to the time, the second term is first order accurate of the time variation of the particle number density and the third term is the overall particle number density change at the k time-step, β and γ are coefficient calibrated on hydrostatic pressure calculations.If negative pressures are obtained after solving this PPE they have to be replaced by zero as they will create higher pressure gradients which increases repulsive force. In general, if the solution of the PPE gives wrong pressure values the result of calculated force will be inaccurate.

v) Two-terms source of the PPE: The modifications on the source term of the Poisson Pressure's equation proposed by Tanaka and Masunaga [31] in 2010 are the following,

$$\nabla^2 P_i^{k+1} = \frac{\rho}{\triangle t}\nabla \cdot u^* + \gamma \frac{\rho}{\triangle t^2}\frac{n^o - n_i^k}{n^o} \tag{2.33}$$

where the first component is the divergence of the temporary velocity and the second one is the variation of the particle number density at the time-step k from the initial particle number density with a relaxing factor γ. Also, they have used the following pressure gradient without any mathematical justification saying that this equation preserves the linear momentum,

$$\nabla P_i^{k+1} = \frac{d}{n^o}\sum_{j\neq i}\left[\frac{P_j + P_i}{\mid \mathbf{r}_j - \mathbf{r}_i \mid^2}(\mathbf{r}_j - \mathbf{r}_i)W(\mid \mathbf{r}_j - \mathbf{r}_i \mid)\right] \tag{2.34}$$

Moreover, this equation was also found to be adopted by Jeong et al [6] and by Lee et al [16] on the model suggested by Toyota et al (2005, in Japanese) in which pressure fluctuations were reduced [16].

vi) Two error compensation terms, ECS: The two error compensations source proposed by Khayyer and Gotoh [10] in 2011 are the following,

$$\left(\frac{\triangle t}{\rho}\nabla^2 P_i^{k+1}\right) = \frac{1}{n^o}\left(\frac{Dn}{Dt}\right)_i^* + \left|\left(\frac{n_i^k - n^o}{n^o}\right)\right|\left[\frac{1}{n^o}\left(\frac{Dn}{Dt}\right)_i^k\right] +$$

$$\left|\frac{\triangle t}{n^o}\left(\frac{Dn}{Dt}\right)_i^k\right|\left[\frac{1}{\triangle t}\left(\frac{n_i^k - n^o}{n^o}\right)\right], n^k > \gamma n^o \tag{2.35}$$

$$\left(\frac{\triangle t}{\rho}\nabla^2 P_i^{k+1}\right) = \frac{1}{n^o}\left(\frac{Dn}{Dt}\right)_i^* + \left|\left(\frac{n_i^k - n_{is}}{n_{is}}\right)\right|\left[\frac{1}{n^o}\left(\frac{Dn}{Dt}\right)_i^k\right] +$$

$$\left|\frac{\triangle t}{n^o}\left(\frac{Dn}{Dt}\right)_i^k\right|\left[\frac{1}{\triangle t}\left(\frac{n_i^k - n_{is}}{n_{is}}\right)\right], n^k \leq \gamma n^o \tag{2.36}$$

where the first term on the right hand side of both equations is the source term proposed by themselves in 2009 [8] equation 2.29 and the followed terms are compensation terms. Also, γ is a smaller constant than the free surface criterion from equation 2.25. They have also proposed the gradient correction GC matrix which is introduced on the gradient model,

$$\nabla P_i^{k+1} = \frac{d}{n^o} \sum_{j \neq i} \left[\frac{P_j + P_i}{| \mathbf{r}_j - \mathbf{r}_i |^2} (\mathbf{r}_j - \mathbf{r}_i) C_{ij} W(| \mathbf{r}_j - \mathbf{r}_i |) \right] \tag{2.37}$$

where,

$$C_{ij} = \begin{pmatrix} \sum_{j \neq i} V_{ij} \frac{W_{ij} x_{ij}^2}{r_{ij}^2} & \sum_{j \neq i} V_{ij} \frac{W_{ij} x_{ij} y_{ij}}{r_{ij}^2} \\ \sum_{j \neq i} V_{ij} \frac{W_{ij} x_{ij} y_{ij}}{r_{ij}^2} & \sum_{j \neq i} V_{ij} \frac{W_{ij} y_{ij}^2}{r_{ij}^2} \end{pmatrix}^{-1} \tag{2.38}$$

and

$$V_{ij} = \frac{1}{\sum_{j \neq i} W_{ij}} \tag{2.39}$$

Further in 2013, Tsuruta, Khayyer and Gotoh [32] clearly stated that this gradient correction did not resolve the maldistribution of particles and it requires a careful setting of calculation conditions. Thus, they proposed a dynamically stabilised, DS, gradient operator as a repulsive force to stabilise the calculation. However, they have not used the DS in further publications as in in Hwang et al [5] in 2014. These papers can bee seen for further details.

As a result of the review of all the above mentioned variants, it can be observed that the main issue is the lack of agreement of researchers to accurately determine the pressure in the implicit Poisson Pressure's equation and the calculation of the pressure gradient. Those variants were implemented and tested on *Python* to determine the performance as it will be shown in Chapter 4. Most of them partially work but there are still pressure oscillations and it can be seen an unphysical particle behaviour. The two terms for the source term proposed by Tanaka and Masunaga [31] and the Corrective matrix proposed by Khayyer and Gotoh [10] did not work.

Chapter 3

Moving Particle Semi-Implicit Method

The Moving Particle Semi-Implicit Method is based on the prediction correction calculation of the velocity field based on the Helmotz-Hodge decomposition. Initially the prediction velocity is calculated based on the viscous and external forces term and then corrected by the gradient of pressure. The pressure is obtained by the solution of the Poisson Pressure's equation. The next algorithm description shows the step by step procedure implemented.

3.1 Algorithm description

1. Particle generation.
   Particles are systematically generated in such a way that an uniform distribution exists over the whole domain including fluid and solid. Hence the initial particle number density n_o is constant as is taken as a reference during the computation. This particle generation over all the domain is done once at the beginning of the simulation. Inlet particles are generated just when it is required according to the fluid flow.

2. Initial conditions setup.

   The fluid velocity is initialised with the free speed flow u_o and initial positions r_o are taken from the particle generation.

3. Neighbour search.

   As particles evolve in time, it is necessary to find the particles in the support domain of a target particle every time step. Three techniques are commonly used (Liu and Liu, [17]).

   - All-pair search: The distance r_{ij} from a particle i to each and every particle j is calculated. This technique is computationally time consuming, reason why it is used for very small scale problems.

   - Linked-list algorithm: The neighbour searching is over a temporary grid overlaid on the domain, where only the same and adjacent cells are looked for particles, i.e. nine cells for a 2D domain and 27 cells for a 3D domain.

   - Tree search algorithm: The problem domain is split into octants that contain particles, until the leaves on the tree are individual particles. The target particle is enclosed in a cube side two times the compact support distance. It is checked if this volume overlaps other volumes in the domain. If there are particles within the support domain those are recorded as neighbours.

   The linked list was used as it was a large particle domain and it is easy to implement every time step.

4. Particle number density calculation.

   The particle number density, pnd, is calculated with equation 2.19

5. Intermediate velocity and position calculation.

   The intermediate velocity is calculated with the viscous term and the external force as,

$$\mathbf{u}^* = \mathbf{u}^k + \triangle t \cdot (\nu \nabla^2 \mathbf{u}^k + \mathbf{g}) \tag{3.1}$$

The intermediate position is calculated with the intermediate velocity as,

$$\mathbf{r}^* = \mathbf{r}^k + \triangle t \cdot \mathbf{u}^* \tag{3.2}$$

6. Poisson Pressure's equation solution.

   The Poisson Pressure equation is implicitly solved using the original equation 2.24 or its variants to obtain the particles pressure. As the system of equations generate a sparse matrix it is solved using a Python library for this kind of matrixes

7. Gradient pressure calculation.

   The Gradient pressure is calculated using the original equation 2.21 or its variants.

8. Velocity correction.

   The velocity correction is calculated with the remainder part of the Navier Stokes' equation which includes the gradient of pressure as,

$$\mathbf{u}' = -\frac{\triangle t}{\rho} \nabla P^{k+1} \tag{3.3}$$

9. Velocity and position update.

   To obtain the new particles velocities and positions, the intermediate values are corrected with the above mentioned velocity as,

$$\mathbf{u}^{k+1} = \mathbf{u}^* + \mathbf{u}' \tag{3.4}$$

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \triangle t \cdot \mathbf{u}^{k+1} \tag{3.5}$$

10. Boundary conditions.

    When a particle is detected as belonging to the free surface a Dirichlet boundary condition is imposed. The PPE will have 1 in its corresponding positions of free surface particles and and zero in the source term. At the inlet, particles are introduced to the domain at upstream velocity and at the outlet particles are removed from the domain. At the bottom there are two layers of particles that moves at free stream velocity. The steps three to nine are repeated until the total time of simulation is reached.

The diagram in figure 3.1 shows the algorithm. Although splitting the velocity in the prediction-correction scheme generates an error, there are not an internal loop between the intermediate and final velocities to minimize the above mentioned error on the MPS method.

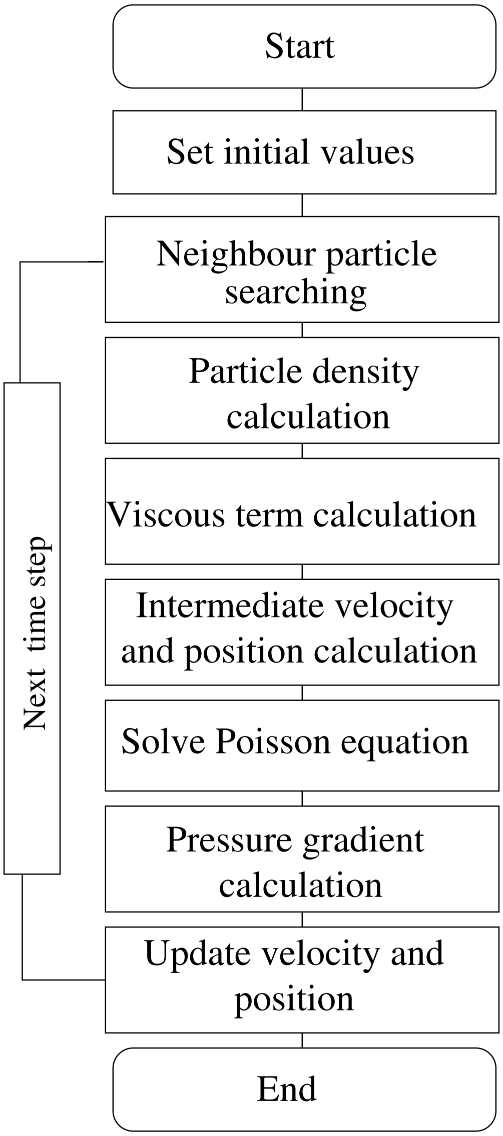

Start
Set initial values
Neighbour particle searching
Particle density calculation
Viscous term calculation
Intermediate velocity and position calculation
Solve Poisson equation
Pressure gradient calculation
Update velocity and position
End
Next time step

Figure 3.1: MPS Algorithm

3.2 Free surface stabilisation

Solving the Poisson Pressure's equation in the original MPS method (equation 2.24) causes negative pressures as the particle number densities near the free surface is smaller than the one for inner particles. Those negative pressure values cause instability [13]. Hence, after the solution of the Poisson's equation if negative pressures are obtained they must be replaced by zero in the original method which is not mentioned in the original MPS papers or any other paper previous to Kondo and Koshizuka [13] as they will create higher pressure gradients which increases repulsive force. In general, if the solution of the PPE gives wrong pressure values the result of calculated force will be inaccurate. Hence, several trials where the physical phenomenon lead to negative pressures always suddenly ended the simulation. On the other hand, gradient calculations over the whole particle domain by the equation 2.21 in the current model of flow around a 2D body creates that the free surface particles rise up leading to a non physical phenomenon. Those problems had been solved in my following proposal.

3.2.1 Poisson Pressure's equation

The mathematical model on the calculation of the Poisson Pressure's equation has been proposed in a similar fashion as in multistep methods. It consist on a linear combination of the source term with information of previous time steps. The first part of the source term is variation of the particle number density respect to the time, discretised by equation 2.29. The second term, is the instantaneous time variation of the particle number density from the previous time step and the third term, is the deviation from the initial particle number density. That is,

$$\frac{\triangle t}{\rho}\nabla^2 P_i^{k+1} = \begin{cases} \alpha\frac{1}{n^o}\left(\frac{Dn}{Dt}\right)_i^* + \beta_1\left|\left(\frac{n_i^k - n^o}{n^o}\right)\right|\left[\frac{1}{n^o}\left(\frac{n_i^k - n_i^{k-1}}{\triangle t}\right)\right] + & \\ \qquad \beta_2\left|\left(\frac{n_i^k - n_i^{k-1}}{n^o}\right)\right|\left[\frac{1}{n^o}\left(\frac{n_i^k - n^o}{\triangle t}\right)\right], & if\ n_i^k > \gamma n^o (3.6a) \\ \alpha\frac{1}{n^o}\left(\frac{Dn}{Dt}\right)_i^* + \beta_1\left|\left(\frac{n_i^k - n_s^o}{n_s^o}\right)\right|\left[\frac{1}{n^o}\left(\frac{n_i^k - n_i^{k-1}}{\triangle t}\right)\right] + & \\ \qquad \beta_2\left|\left(\frac{n_i^k - n_i^{k-1}}{n_s^o}\right)\right|\left[\frac{1}{n^o}\left(\frac{n_i^k - n_s^o}{\triangle t}\right)\right], & if\ n_i^k \leq \gamma n^o (3.6b) \end{cases} \tag{3.6}$$

where α, β_1 and β_2 are relaxation factors, n^o and n_s^o are the initial particle number density of a particle inside the domain and a particle located on the free surface respectively, γ can be the same free surface criterion β from equation 2.25 .

When the fluid is being compressed the instantaneous variation of the particle number density is positive, on the contrary when the fluid is being expanded the instantaneous variation is negative. In the same way, when the fluid has already been compressed the deviation from the initial particle number density is positive, on the contrary when the fluid has already being expanded the deviation is negative. In this way when the fluid is being expanded and it has already been compressed the source term leads just to the equation 2.29. Equation 3.6 a) is used for inner particles and equation 3.6 b) is for free surface particles.

3.2.2 Gradient condition

As the particle number density near to the free surface tends to be smaller than the one of inner particles, the source term and pressure values after solving the Poisson Pressure equation are calculated as negative ones in the traditional MPS which causes instabilities. Hence, those negative pressures must set to zero in order to avoid that issue [13]. This was clearly experienced when negative pressures were allowed causing particle disorders and the simulation crashes in early stages of the computation. The following condition has been imposed over the particles to calculate the gradient,

$$n_i > \beta n^o \tag{3.7}$$

with β equals to 0.95, otherwise the gradient is zero. This means, particles near to the free surface do not have gradient pressure and it was taken only on inner particles. This condition improved considerably the fluid flow behaviour meanwhile in the original MPS gradient was calculated over the whole domain and particles on the free surface tended to rise up.

These two contributions were implemented in *Python 3.2* with good results and experimental visualisation has reinforced the numerical data. Additionally, there are no reports on the literature so far which mention the estimation of drag resistance using the Moving Particle Semi-Implicit method. This could be one of the first on that field.

Chapter 4

Numerical Examples

4.1 Dam break

The dam break is a common benchmark used to validate computations with free surface due to its quick evolution of fluid walls in time. It consists on a fluid column initially at rest that collapses, hence the height of the column falls and the fluid spreads out. In this case, the dam break was tested for 648 fluid particles, the same number as the original MPS method proposed by Koshizuka and Oka [15]. The initial configuration is a rectangular particle array of a=0.216 meters width by twice a, 2a=0.432 meters height. Also, the domain width is 4 times a, 4a =0.864 meters, and the domain height is just important to keep particles inside it. Particles are separated by a distance from their centres by 0.012 meters and the time step used was 1×10^{-3}seconds. The main parameters and the initial configuration are shown in table 4.1 and figure 4.1 respectively. The equations used here were the original MPS ones and the condition of zero pressure was imposed to free surface particles which reached the condition of the equation 2.25.

Parameter	Value
Initial particle spacing	lo=0.012 m
Time step	1×10^{-3} sec
Free surface criteria	$\beta = 0.97$
Fluid density	1000 Kg$/m^3$
Kinematic viscosity	1×10^{-6} m^2/s

Table 4.1: Calculation conditions for the dam break.

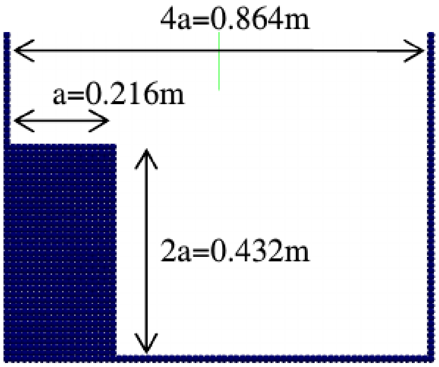


Figure 4.1: Initial configuration of the dam break

The figure 4.2 shows snapshots of particles performance at 0.5 and 1.0 seconds. At a glance, it can be seen how particles of the current simulation on the left hand side behave in similar way as compared with Kondo and Koshizuka [13] on the right hand side.

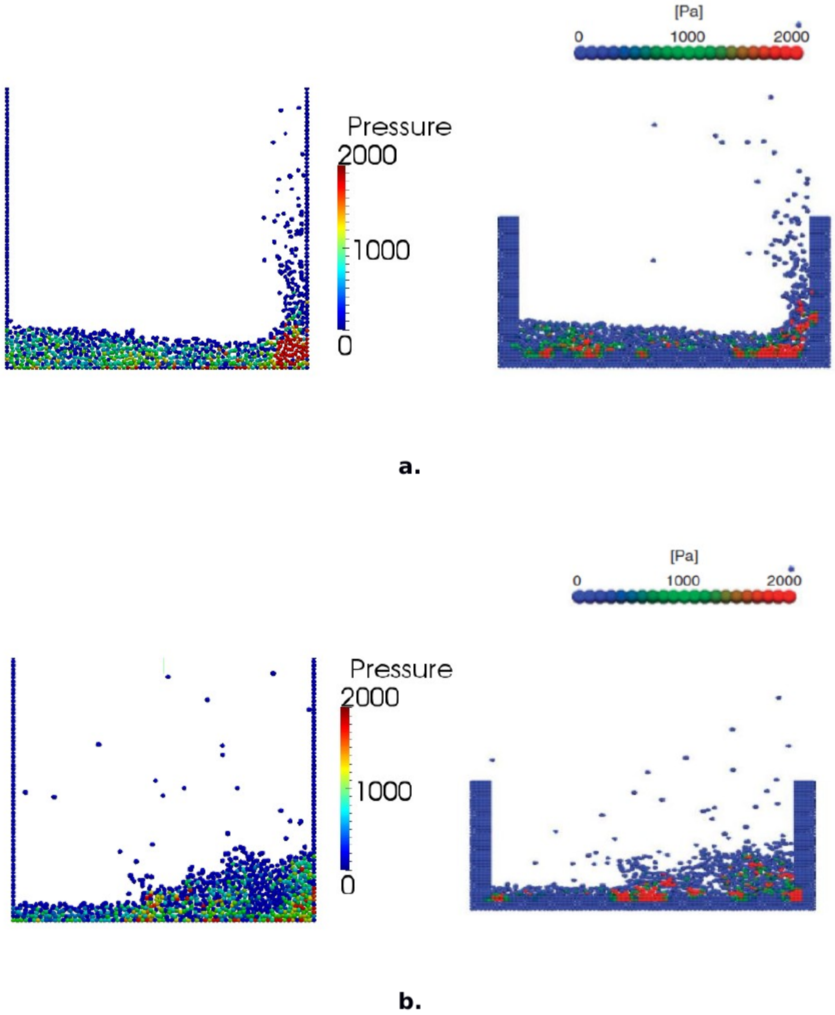


Figure 4.2: Snapshots of the dam break at a) t=0.5 and b) 1.0 seconds respectively. Left: current simulation. Right: Kondo and Koshizuka [13]

The advance of the right bottom end of the column of water was tracked. In order to compare with data on the literature it was convenient to present the results in units of the non-dimensional quantities. Figure 4.3 shows the evolution of the non-dimensional right bottom end of the column compared with experimental data from Moyce [23]. It can be said that computational results are in good agreement with experimental ones. Hence the MPS method was verified and validated for a closed domain. In the following, particles are required to enter and leave the domain in order to simulate a fluid flow around a two dimensional body.

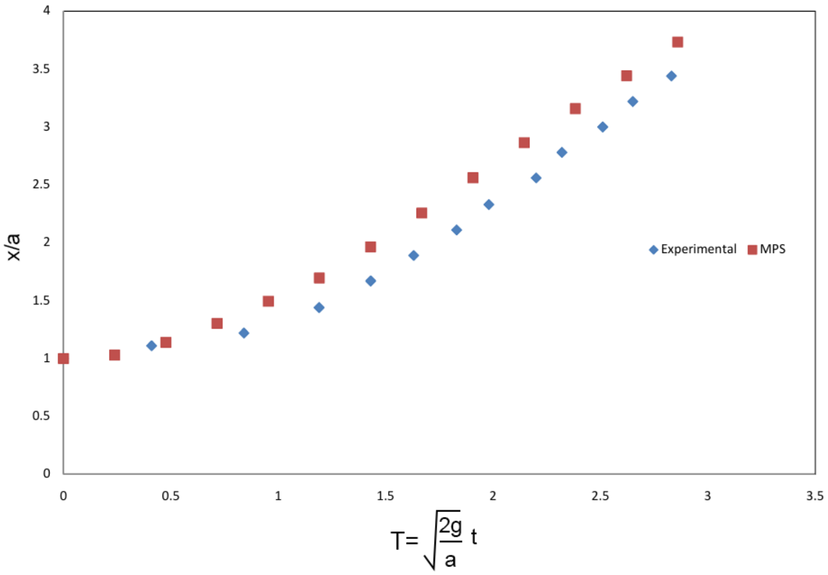


Figure 4.3: Non-dimensional edge of advance of the dam break test versus dimensionless time. Comparison of MPS with experimental results. Where, a: width of the liquid column base, x: horizontal edge of advance of the water [23]

4.2 Flow around a 2D square

A solid square with side length L was introduced into a rectangular fluid flow particle domain as it can be seen in Figure 4.4The solid domain was modelled as a set of particles. Fluid particles were allowed to leave the domain when they reached the outlet on the left end and new fluid particles were created and introduced on the right side in order to keep the fluid flow continuously.

It was observed that the simulation exhibited an unstable behaviour using the original MPS method. This means that particles suddenly get out of the domain. In order to stabilise the simulation the following contributions, described in section

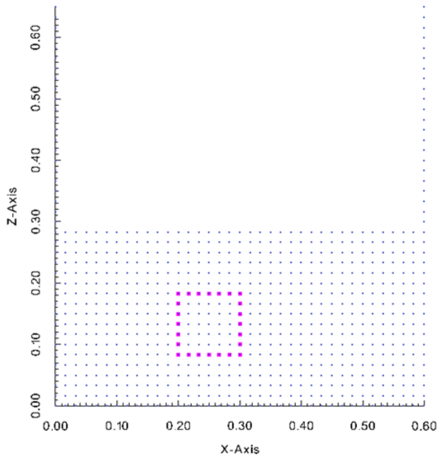


Figure 4.4: Fluid particles domain. The red particles correspond to the boundary of the solid square. Dimensions are given in meters.

2.2.7, where implemented and tested:

CMPS The Corrected MPS.

HS The High Source term for the Poisson Pressure Equation.

HL The High Laplacian

HV The High Viscosity

ECS Error Compensation Source term.

Other modifications were tested as the two components of the source term proposed by Tanaka and Masunaga [31] and the Corrective matrix proposed by Khayyer and Gotoh [10] but they did not work.

The following is the summary of MPS models, ending with better results given by the scenario 3

	MPS Model	Observation
Scenario 1	CMPS, HS	Particles penetrate domain boundaries
Scenario 2	CMPS, HS, HL, HV	Particles leave suddenly the domain
Scenario 3	CMPS, HS, HL, HV and ECS	Better results but there are still the mentioned issues

Based on the previous results a MPS model, with the corrections CMPS, HS, HL, HV and ECS, was used test the influence of the number of particles and the solid size in the simulation. The following cases were evaluated:

	MPS Model	Solid depth	Particles	Solid Size
Case 1	CMPS, HS, HL, HV and ECS	0.05m	369	small
Case 2	CMPS, HS, HL, HV and ECS	0.15m	579	small
Case 3	CMPS, HS, HL, HV and ECS	0.10m	546	2x

4.2.1 Case 1

Two different time steps 1×10^{-3} and 1×10^{-4} seconds were tested for a initial particle spacing lo$=$ 0.0166m, cut-off radius $re = 2.1$ lo, total number of 369 particles, velocity inlet $v = 2.081$ m/s, $L = 3$ $l_o$$=$ 50×10^{-3}m and a given Reynolds number Re $\approx$ 100.000. Negative pressures obtained by solving the Poisson Pressure's equation were set to zero, otherwise simulation presented stability issues, showing that particles get out of the domain and penetrate the solid, as a consequence the simulation ended at few time steps. As it can be seen in figure 4.5 both simulations presented solid penetration. For the larger time-step, 1×10^{-3} seconds, the simulation run 495 milliseconds before particles suddenly get out of the domain, nonetheless the fluid was not able to fall behind the square. For the smaller time-step, 1×10^{-4}, the simulation run 189 ms before suddenly get out of the domain.

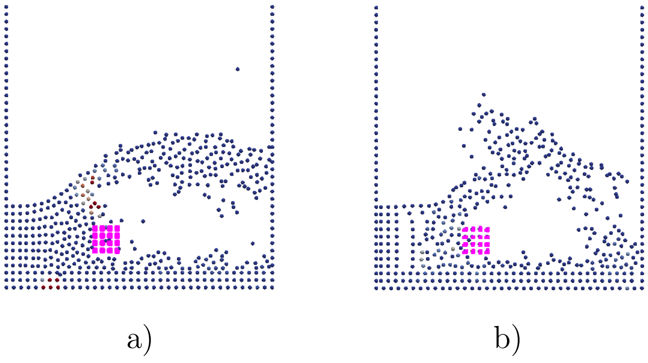


Figure 4.5: Case 1: Flow around a 2D small square and 369 fluid particles a) time-step =1×10^{-3}, time =495 milliseconds b) time-step =1×10^{-4} , time =189 milliseconds, CMPS, HS, HL, HV, ECS.

Particles at the inlet of the flow must keep the initial distribution and hence preserve the same particle number density as the initial one n_o. The variation in time of density must be zero other wise the mass conservation is not satisfied. As it can be seen in figure 4.5 there is a clear violation of fluid incompressibility as particles near to the inlet do not keep its initial distribution.

The drag resistance coefficient is calculated every time step according to the formulae $C_d = 2F/\rho V^2 A$. Figure 4.6 plots C_d for the two time-steps tested 1×10^{-3} and 1×10^{-4} seconds. Both coefficients overlaps until 90 milliseconds when the small time-step 1×10^{-4} increases the pressure fluctuations and the simulation ends earlier at 189 milliseconds. The average of the drag resistance calculated over all the time-steps is reported as $\overline{C_d}$ =0.47 for time-step =1×10^{-3} and $\overline{C_d}$ =0.62 for time-step 1×10^{-4}.

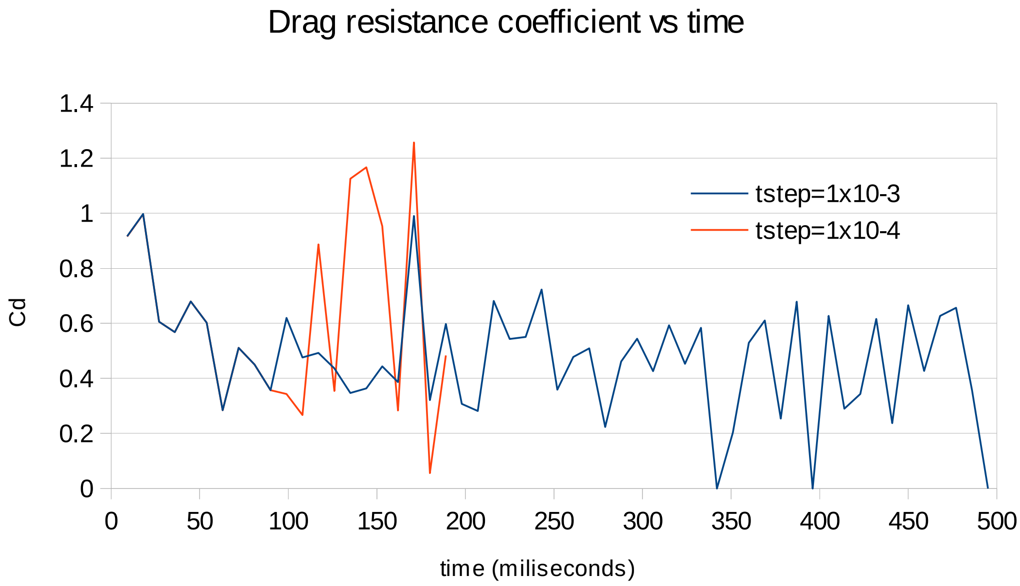


Figure 4.6: Drag resistance coefficient for Case 1, a 2D small square and 369 fluid particles a) time-step =1×10^{-3} , $\overline{C_d}$ =0.47, total time =495 ms b) time-step = 1×10^{-4}, $\overline{C_d}$ =0.62, total time =189 ms, CMPS, HS, HL, HV, ECS.

4.2.2 Case 2

This numerical test uses the same model from the last example but the number of fluid particles was increased to 579, also increasing the fluid depth, i.e. the solid is at deeper water, and the inlet velocity was reduced to the half of the previous simulation to v=1.0405 m/s, solid side $=3lo = 50\times10^{-3}$m, given a Reynolds number Re ≈50.000. For the larger 1×10^{-3} seconds, the simulation run 873 milliseconds, almost twice the time than before but it still the particles get out the domain. There are solid penetration and violation of fluid incompressibility since particles are separated from each showing spaces as it can be seen in figure 4.7. For the smaller time-step, 1×10^{-4} seconds, the simulation run 54 milliseconds before it blows up. The drag resistance coefficient variation in time is reported in figure 4.8. The average drag coefficient was calculated according to the formulae $C_d = 2F/\rho V^2 A$.

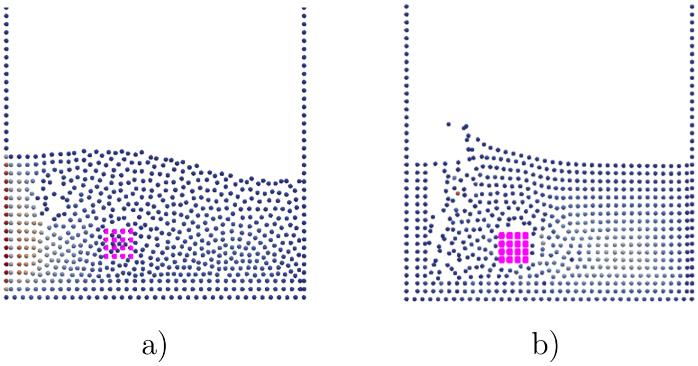


Figure 4.7: Case 2: Flow around a 2D small square and 579 fluid particles a) time-step $=1 \times 10^{-3}$, time =873 ms b) time-step $=1 \times 10^{-4}$, time =54 ms, CMPS, HS, HL, HV, ECS.

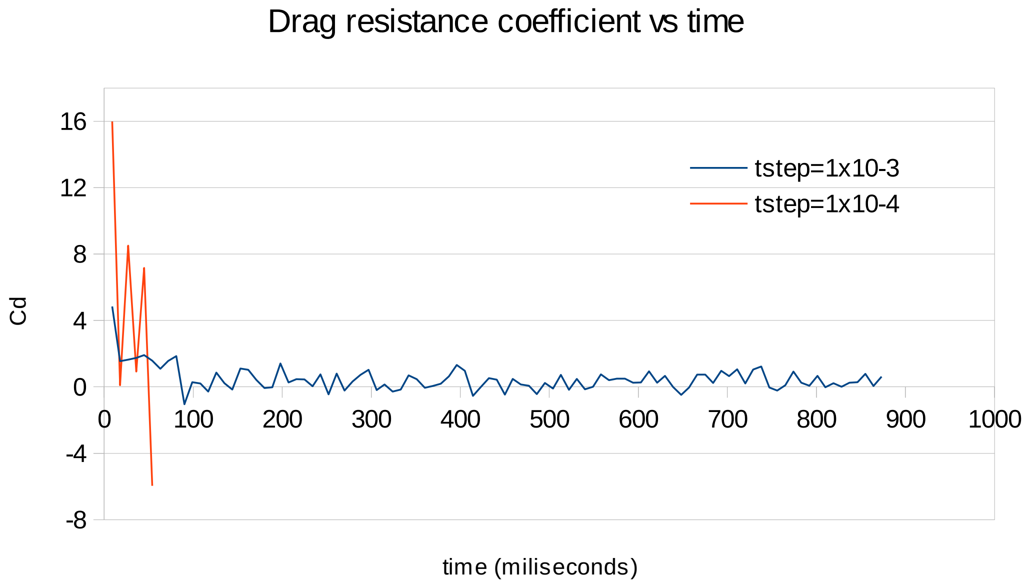


Figure 4.8: Drag resistance coefficient for case 2, a 2D small square and 579 fluid particles a) time-step $=1 \times 10^{-3}$, $\overline{C_d}$ =0.47, total time =873 ms b) time-step $= 1 \times 10^{-4}$, $\overline{C_d}$ =4.95, total time =54 ms, CMPS, HS, HL, HV, ECS.

Comparing the same scenario with different number of particles, higher number of particles generate more instabilities for the smallest time step. This means that more particles are prone to bump against each other creating big repulsions. That is the reason why in the second case, of 579 particles, the simulation ends at 54 milliseconds compared with first case, of 369 particles that ends at 189 milliseconds for 1×10^{-4}seconds. On the other hand, increasing the number of particles keeps the simulation more stable for lower time step but there is still solid penetration.

4.2.3 Case 3

The solid object was increased to a double size and the velocity was kept as in the previous simulation to v=1.0405 m/s, $L = 6\,lo = 100 \times 10^{-3}$m, given a Reynolds number Re ≈100.000. For the larger time-step, 1×10^{-3} seconds, simulation just run 72 milliseconds and 63 milliseconds for the smaller time-step, 1×10^{-4} seconds. Results can be seen in figure 4.9. The drag resistance coefficient variation in time is reported in figure 4.10. The average drag coefficients was calculated according to the formula $C_d = 2F/\rho V^2 A$. The biggest pressures have been observed with the smallest time-steps.

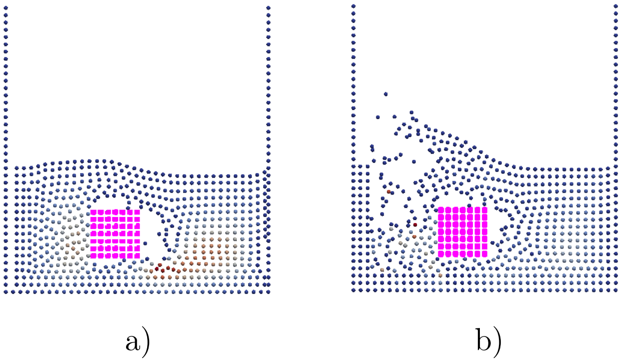


Figure 4.9: Case 3: Flow around a 2D bigger square and 546 fluid particles a) time-step $=1 \times 10^{-3}$, time $=$72 ms b) time-step $=1 \times 10^{-4}$ time $=$63 ms, CMPS, HS, HL, HV, ECS.

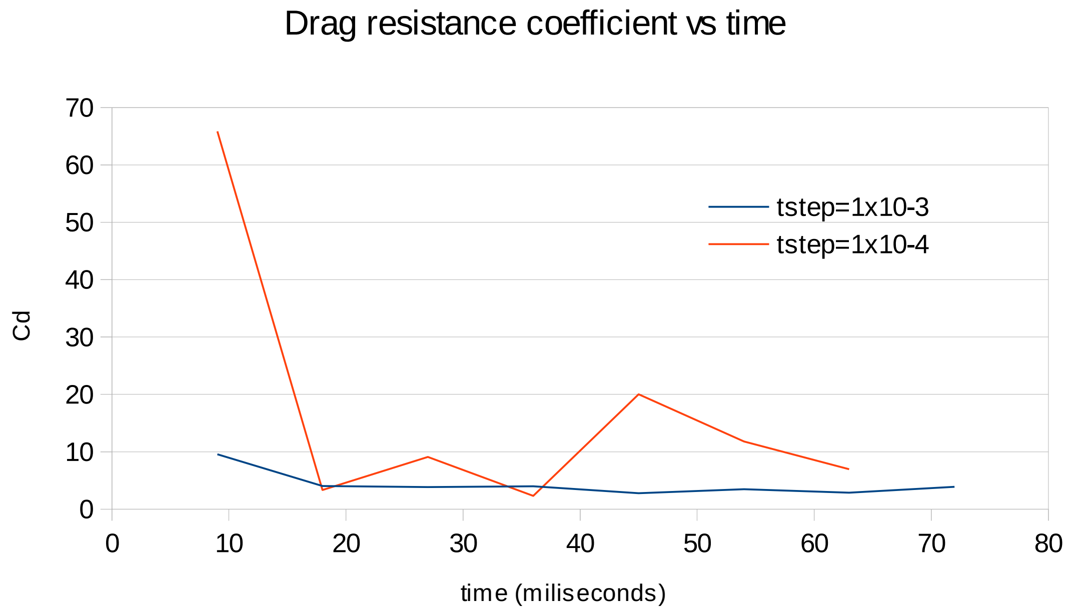


Figure 4.10: Drag resistance coefficient for Case 3, a 2D square and 546 fluid particles a) time-step $= 1 \times 10^{-3}$, $\overline{C_d}$ =4.31, total time =72ms b) time-step $= 1 \times 10^{-4}$, $\overline{C_d}$ =17.06, total time =63 ms, CMPS, HS, HL, HV, ECS.

As it can be observed from the experiments presented in this section, large time steps, of the order of 1×10^{-3}s give rise to solid penetration of the fluid particles when the solid is not too dense. On the other hand, time steps, of the order of 1×10^{-4}s result in unstable and wrong solutions.

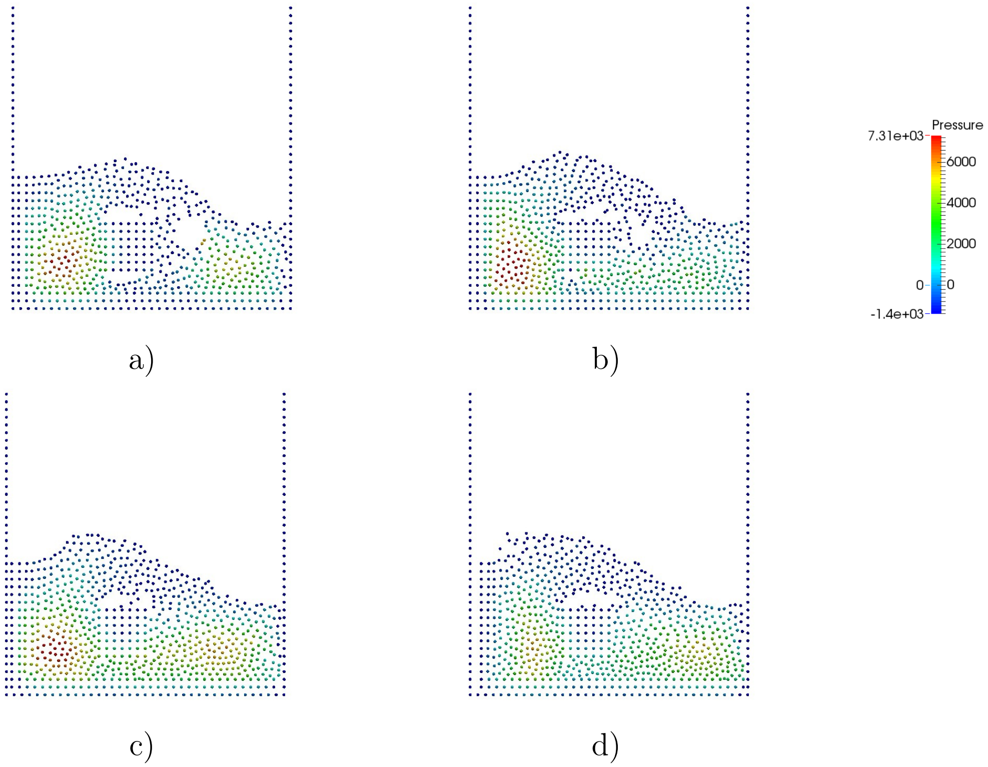
The Courant-Friedrichs-Lewy (CFL) condition was implemented and tested as a conventional way for convergence in many time marching simulations. The CFL condition defines the time step as a function of the velocity, time step and length of the interval, in this case the spatial length is defined by initial particle separation. Nonetheless, as time goes, due to errors associated with calculation of speed, these values are increased and the time step decreases. This turns the simulation unstable when calculating the PPE by the dependence from the quadratic time-step in the original MPS method. When the algorithm included the CFL condition velocities increased every time step and the simulation crashed.

4.2.4 Free surface stabilisation model test

The model proposed on the section 3.2 was implemented initially for a set of 546 fluid particles. .The same initial conditions as in the previous scenario were tested, time steps of 1×10^{-4} seconds, initial particle distance $lo=$ 0.0166m, cut-off radius re = 2.1 lo, fluid particles =546, velocity inlet v=1.0405 m/s, L = 100×10^{-3}m and a given Reynolds number Re $\approx$ 100.000. The relaxation factors for the equations 2.35 and 2.36 were selected as $\alpha = 0.1$ and $\beta_1 = \beta_2 = 0.01$. Figure 4.11 shows the particle evolution in time coloured by pressure values.

It can be seen that the pressure distribution has regions into the whole domain with no local peak values. Colour pressure patches are clearly distinguished. The free surface has smooth transition and there is no evidence of violation of fluid incompressibility.

This scenario was not compared with other computations which use the original MPS, as it was not found literature that reports this experiment. For this reason experimental validation of this example was developed. The results are shown in section 4.3. The drag resistance coefficient advancing in time is reported in figure 4.12. The average value of this coefficient calculated according to the formulae $C_d = 2F/\rho V^2 A$ is $C_d = 2.1$ which is in a good agreement with the value reported by the literature [33] $C_d \approx 2.15$ for rectangular cylinders at Reynolds number in a range from 1×10^4 to 1×10^5.



a)

b)

c)

d)

Figure 4.11: Particle evolution in time of the flow around the 2D square, 546 fluid particles, time-step $=1\times10^{-4}$ seconds, pressure coloured. a) 250ms b) 500ms c)750ms d)1second

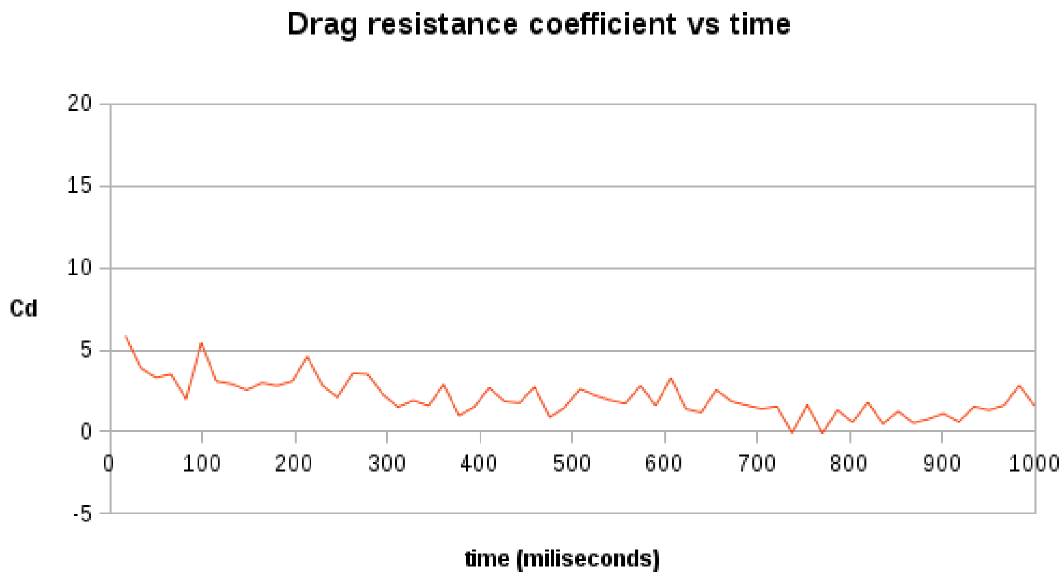


Figure 4.12: Drag resistance coefficient of the 2D square versus time with proposed modifications. time-step = 1×10^{-4} seconds

With this variant of the MPS, the simulation is quite stable and the fluid flow particles exit the domain in a controlled maner. After carried out this simulation with 546 fluid particles, in a second experiment, this number was increased to 2281, l_o was reduced to 0.0083m, the cut-off radius was maintained at $re = 2.1$ lo, velocity inlet $v = 1.0405$ m/s, $L = 100x10^{-3}$ m and a given Reynolds number Re ≈ 100.000. Figure 4.13 shows a better behaviour of the fluid flow. This same behaviour has been ploted with darken particles declared as free surface as it can be seen in figure 4.14

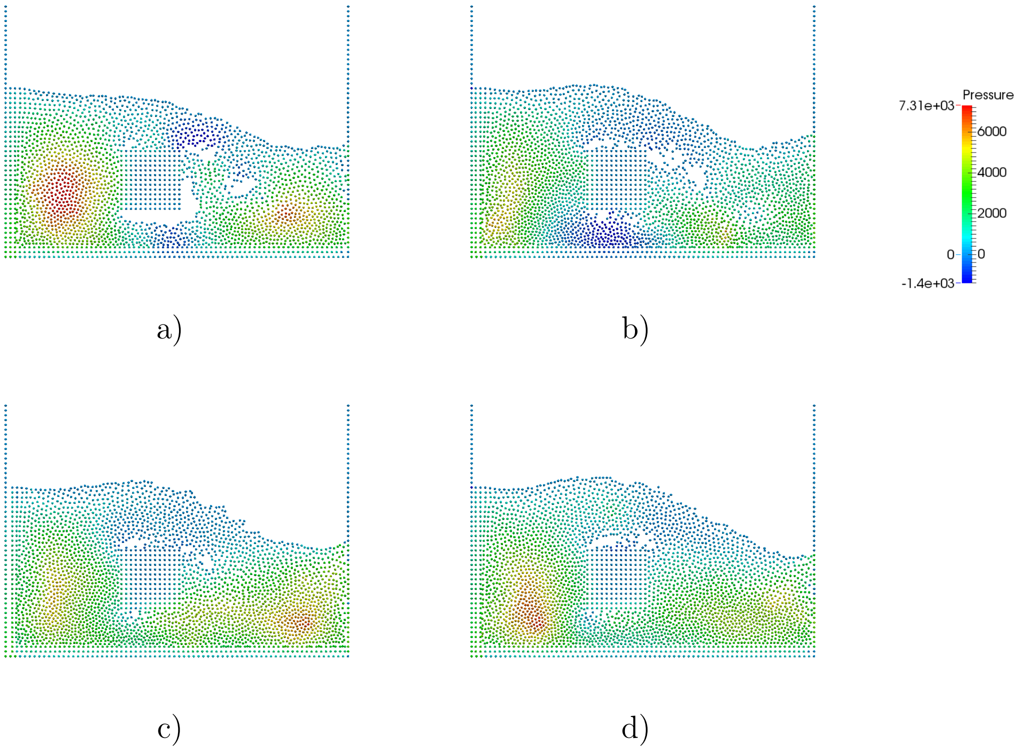
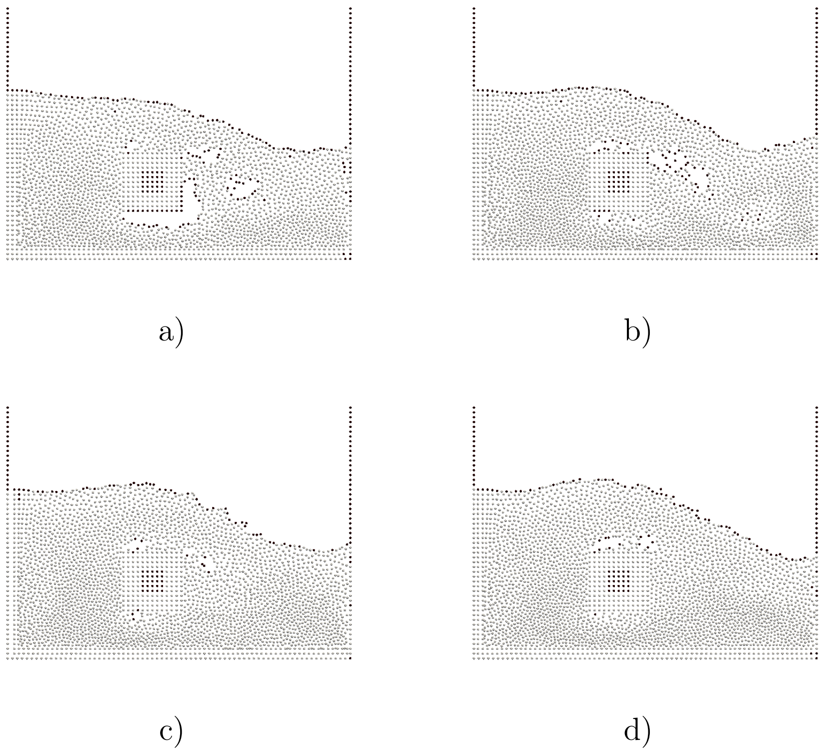


Figure 4.13: Particle evolution in time of the flow around the 2D square, 2281 fluid particles, time-step = 1×10^{-4} seconds, pressure coloured. a) 250ms b) 500ms c)750ms d)1second



a) b)

c) d)

Figure 4.14: Particle evolution in time of the flow around the 2D square, 2281 fluid particles, time-step $=1 \times 10^{-4}$ seconds. Particles that meet the free surface criteria are darken. a) 250ms b) 500ms c)750ms d)1second

4.3 Experimental validation

In order to validate the computational data, a squared cylinder was assembled on an open channel flow. The dimension of the base of the squared cylinder was 10 cm by 10 cm, the same dimension as in the simulations. The dimension of the channel was 30 cm width by 50 cm high by 4.5 m long. The maximum flow capacity of the channel was 28 lts/s. Figure 4.15 shows the assembly the squared cylinder in the open channel. The lateral view of the squared cylinder can be seen in figure 4.16.

The channel was filled of water until the desired free surface level with an average velocity inlet of 1 m/s. This velocity was calculated based on the volumetric flow

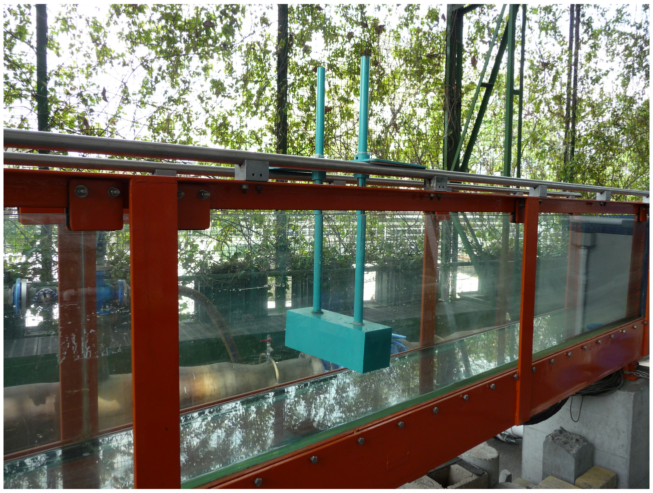


Figure 4.15: A squared cylinder in the open channel used to validate the simmulations

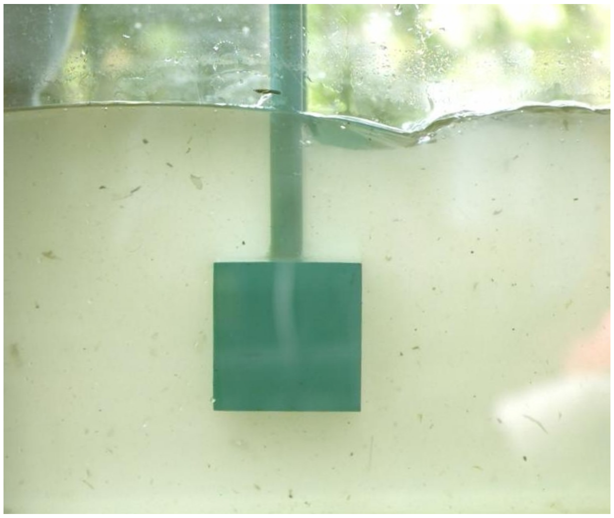
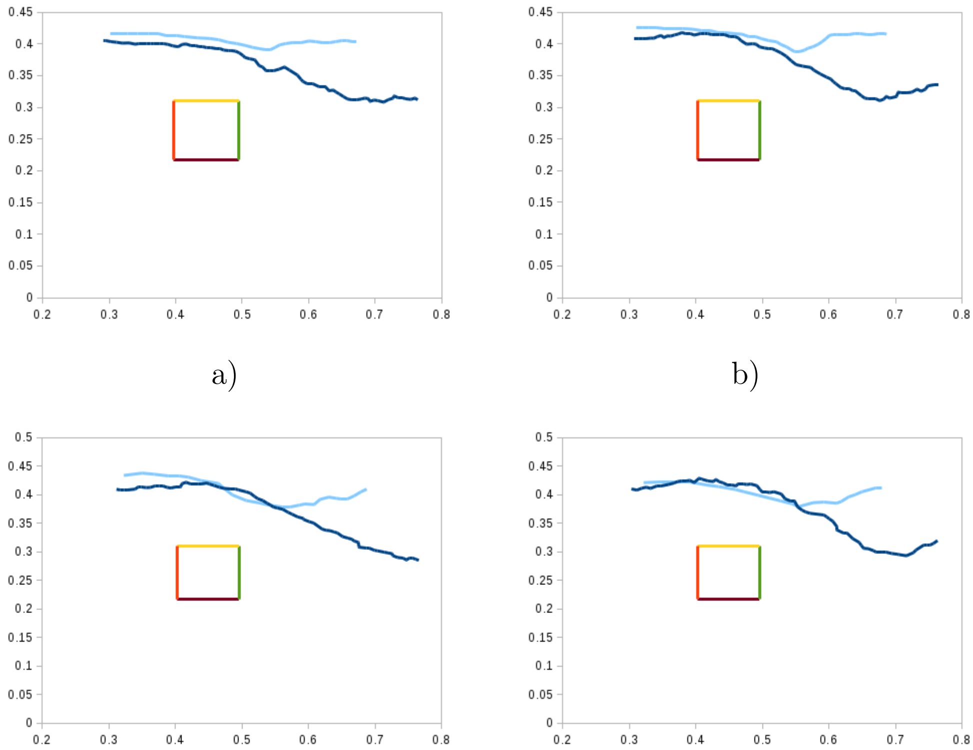


Figure 4.16: Square cylinder used for experimental validation

measurements in the pipe that provides water to the channel. Then, the squared cylinder was immersed and few seconds afterwards the evolution of the free surface was recorder with a high speed, high resolution camera. As mentioned above, the square was built with the same dimensions as in the computations and the flow speed was set at 1.0 m/s. Nevertheless, the experiment differ as the cylinder has a couple of vertical guides to move the cylinder in place. The images where procesed in order to recover the free surface at different times. The results can be seen in figure 4.17. The cyan, or light blue, is the experimental free surface and the navy, or dark blue, is the computational free surface. It can be observed that the computed surface on top of the solid is in agreement with the experimental free surface. However, it can be observed that after the flow passes the square the computational free surface falls down while the experimental surface goes up again reaching the hight of the initial free surface.



a) b)

c) d)

Figure 4.17: Free surface comparison. The cyan or light blue is the experimental and the navy or dark blue is the computational free surface. a) 250ms b) 500ms c)750ms d)1second

Differences in the results can be explained by the differences between the the experimental and computational domain: the experiment was carried out in a 3D channel flow meanwhile the computational simulation was done considering just two dimensions. The open channel influences the flow behaviour as the computational domain was smaller than the experimental one. The boundary conditions were imposed to simulate the extension of the channel, however the supporting structure of the cylinder was not taken into account in the simulation. The experimental channel was closed at the end so a drop in the level of the channel was controled. Hence, it can be said that the second half of the free surface differ due to the outlet condition.

In the simulation, when particles reached the left hand side of the domain they were simply deleted. In the water channel, there is a static pressure of the rest of the fluid flow which causes a damming of flow downstream. The cylinder squared was restricted on the side direction so the drag resistance was not measured.

Conclusions

This thesis had studied the behaviour of the flow around a two dimensional body with free surface using the Moving Particle Semi-implicit method, MPS. Modifications of the original MPS found in the literature were implemented and a new proposal of free surface stabilisation was carried out in Python 3.2. Experimental data was obtained with a prototype of the 2D model in an open channel water flow and results were compared.

The original Moving Particle Semi-Implicit method has serious issues with pressure oscillations and also negative pressures are not allowed. One of the main problems is the solid penetration and computational instabilities which causes particles to get apart one from each other as time goes and the simulation suddenly crashes. Several variants of the method were studied and implemented as those proposed by Khayyer and Gotoh, Kondo and Koshizuka, Tanaka and Masunaga. Nonetheless, the simulation of the flow around a 2 D body still presented instabilities.

Regarding to the time step, it was observed how the MPS and its variants presented the following issue: when the time step was as high as 1×10^{-3} seconds, solid penetration was observed as the simulation did not capture the right physics. On the other hand, when the time step was as low as 1×10^{-4} seconds, particles just blow up and simulations crashed. This was correctly solved implementing the above mentioned modifications and the final simulations were run with a time step of 1×10^{-4} seconds without any problem at all.

This thesis proposed a model for the flow around a 2D object where the pressure given by the solution of the Poisson Pressure's equation can take negative values. Hence, it is not necessary to up date those values to zero as it happened with the original MPS. First, in the source term of the Poisson Pressure's equation has been conceived as a linear combination of particle number densities of previous steps and the initial set up, also there is a different treatment if particles belong to the free surface or not. This gives more computational stability.

The second proposal is a the implementation of a condition over the gradient where the gradient pressure is only calculated for inner particles and not for particles belonging to the free surface. Hence, the particles and the whole fluid flow seems to obey to a physical behaviour.

Increasing the number of particles shows better fluid behavior as it can be seen after implementing the proposed model for 546 and then for 2281 fluid particles. The pressure distribution is clearly distinguished by uniform colour patches or well identified zones. Also, there is a soft transition between pressure values in the fluid and particles with meet the free surface criteria belong indeed to a free surface.

The averaged computational drag coefficient for a 2D squared of 2.1 obtained was in a good agreement with the literature values for a Reynolds number of 1x10^5. The experimental value for the drag coefficient with free surface reported by Venougopal is the same for completely immersed bodies.

Experimental validation of the free surface behaviour was done with a rectangular cylinder, where the first half of the compared computational and experimental free surface were in good agreement. It can be thought that the second half differ due to the outlet condition. In the simulation, when particles reached the left hand side of the domain they were simply deleted. In the water channel, there is a static pressure of the rest of the fluid flow which causes a damming of flow downstream. There are differences between experimental conditions and computations, however good results can be achieved reproducing part of the free surface.

Further work and recommendations

I truly encourage the readers of this thesis to go deep in the knowledge of the Moving Particle Semi-Implicit method. At the beginning of this research no one could have foreseen the issues that were presented here. Some ideas are regarding to the study and implementation Hamiltonian methods for the mathematical model and Gaussian Random Fields for finding neighbours. Applications on Digital Entretainment Engineering is a quite promising field for this sort of particle methods. So go on and move deeper into MPS.

Publications

The following have been the publications regarding to this research so far:

- Flow behaviour over a 2D body with free surface usign a modified moving particle semi-implicit method. Proceedings of The Virtual Concept International Workshop On Green Technologies And Agro-Mechanical Engineering. ISBN 978-2-9548927-2-6. October 7-8, 2015, Bucaramanga Colombia.

An extension of this paper has been approved to publish to the International Journal on Interactive Desing and Manufacturing, edited by Springer under the identification numbers ISSN 1955-2513 (print version) and ISSN 1955-2505 (electronic version).

- Drag resistance over a 2d square using the MPS method. Proceedings of the 11th World Congress On Computational Mechanics. XI WCCM 2014, July 20 - 25, 2014, Barcelona, Spain.

- Implementation of the MPS method for wave tank simulations. Proceedings of the IX Congreso Colombiano De Metodos Numericos: Simulación en Ciencias y Aplicaciones. IX CCMN 2013, Agosto 21-23, 2013, Cali, Colombia.

- Added resistance of a monohull in regular head waves using numerical simulations with mesh movement. Proceedings of the 10th World Congress On Computational Mechanics. X WCCM 2012, July 08-13, 2012. Sao Paulo, Brasil.

Bibliography

[1] B. Ataie-Ashtiani and L. Farhadi. A stable moving particle semi-implicit method for free surface flows. *Fluid Dynamics Research*, 38:241–256, 2006.

[2] R. Azcueta. Ranse simulations for sailing yachts including dynamic sinkage & trim and unsteady motions in waves. *In Proceedings of High Performance Yacht Design Conference, Auckland, December 4-6.*, 2002.

[3] 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.

[4] M. Gomez-Gesteira, B. Rogers, R. Dalyrmple, and A. Crespo. State of the art of classical sph for free-surface flows. *Journal of Hydraulic Research*, 48:6–27, 2010.

[5] A. G.-H. a. P.-J. Hwang, S.C.and Khayyer. Development of a fully lagrangian mps-base coupled method for simulation of fluid-structure interaction problems. *Journal of Fluids and Structures*, 50:497–511, 2014.

[6] S.-M. Jeong and S.-C. P. J.-C. K. M.-H. Nam, J-W.and Hwang. Numerical prediction of oil amount leaked from a damaged tank using two dimensional moving particle simulation method. *Ocean Engineering*, 69:70–78, 2013.

[7] A. Khayyer and H. Gotoh. Development of cmps method for accurate water-surface tracking in breaking waves. *Coastal Engineering Journal*, 50(2):179–207, 2008.

[8] A. Khayyer and H. Gotoh. Modified moving particle semi-implicit methods for the prediction of 2d wave impact pressure. *Coastal Engineering*, 56(4):419–440, 2009.

[9] A. Khayyer and H. Gotoh. A high order laplacian model for enhancement and stabilization of pressure calculation by mps method. *Applied Ocean Research*, 32:124–131, 2010.

[10] A. Khayyer and H. Gotoh. Enhancement of stability and accuracy of the moving particle semi-implicit method. *Journal of Computational Physics*, 230(8):3093–3118, 2011.

[11] A. Khayyer and H. Gotoh. A 3d high order laplacian model for enhancement and stabilization of prssure calculation in 3d mps based simulations. *Applied Ocean Research*, 2012.

[12] A. Khayyer and H. Gotoh. Enhancement of performance and stability of mps mesh-free particle method for multiphase flows characterized by high density ratios. *Journal of Computational Physics*, 242:211–233, 2013.

[13] M. Kondo and S. Koshizuka. Improvement of stability in moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, 65:638–654, 2011.

[14] S. Koshizuka, A. Nobe, and Y. Oka. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, 26(7):751–769, 1998.

[15] S. Koshizuka and Y. Oka. Moving particle semi-implicit method or fragmentation of incomprssible fluid. *Nuclear Science and Engineering*, 123:421–434, 1996.

[16] B. Lee, J. Park, M. Kim, and S. Hwang. Step by step improvement of mps method in simulating violent fre-surface motions and impact-loads. *Computer Methods in Applied Mechanics and Engineering*, 200(9):1113–1125, 2011.

[17] S. Li and W. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55(1):1–34, 2002.

[18] L. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82(12):1013–1024, 1977.

[19] J. J. Monaghan. Why particle methods work. society for industrial and applied mathematics. *Journal of Science on Scientific Computing,*, 3(4):422–433, 1982.

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

[21] J. J. Monaghan. Simulating free surface flows with sph. *Journal of Computational Physics*, 110:399–406, 1994.

[22] J. J. Monaghan. Theory and applications of smoother particle hydrodynamics. *In Blowey J.F. & Craig A.W. (Eds) Frontiers in Numerical Analysis.Netherlands: Springer-Verlag.*, pages 143–193, 2005.

[23] W. Moyce and M. J.C. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London A*, pages 312–325, March 1952.

[24] C. A. Perez-Gutierrez. Hydrodynamic added resistance prediction of a hull in regular head waves using a commercial ranse software. Master's thesis, University of Southampton, 2010.

[25] S. Shao and E. Lo. Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface. *Advanced Water Resources*, 26:787–800., 2003.

[26] K. Shibata and S. Koshizuka. Numerical analysis of shipping water impact on a deck using a particle method. *Ocean Engineering*, 34:585–593, 2007.

[27] K. Shibata, S. Koshizuka, M. Sakai, and K. Tanizawa. Lagrangian simulations of ship-wave interactions in rough seas. *Ocean Engineering*, 42:13–25, 2012.

[28] S. Shibata, K.and Koshizuka and T. K. Three-dimensional numerical analysis of shipping water onto a moving ship using a particle method. *ournal of Marine Science and Technology*, 14:214–227, 2009.

[29] M. Sueyoshi, M.and Kashiwagi and S. Naito. Numerical simulation of wave-induced nonlinear motions of a two dimensional floating body by the moving particle semi-implicit method. *Journal of Marince Science and Technology*, 13:85–94, 2008.

[30] T. Tamai and S. Koshizuka. Least squares moving particle semi-impliciti method. *Computational Particle Mechanics*, 1(3):277–305, 2014.

[31] M. Tanaka and T. Masunaga. Stabilization and smoothing pressure in mps method by quasi-compressibility. *Journal of Computational Physics*, 229(11):4279–4290, 2010.

[32] A. Tsuruta, N.and Khayyer and H. Gotoh. A short note on dynamic stabililzation of moving-particle semi-implicit method. *Computers & Fluids*, 82:158–164, 2013.

[33] V. Venugopal, K. S. Varyanib, and N. D. Barltrop. Wave force coefficients for horizontally submerged rectangular cylinders. *Ocean Engineering*, 30:1669–1704, 2006.

General References

1. Arfken G.B. & Weber H.J. (2005). *Mathematical Methods for Physicists*. Sixth Edition.

2. Chorin A. & Marsden J.E. (1992). *A Mathematical Introduction to Fluid Mechanics*. Third Edition.

3. Ferziger J.H. & Peric M. (2002). *Computational Methods for Fluid Dynamics*.

4. Liu G. & Liu M. (2003). *Smoothed Particle Hydrodynamics a Meshfree Particle Method*. Singapore: World Scientific Publishing Co.

5. Tupper E. (2004). *Introduction to Naval Architecture*. Oxford: Elsevier.

6. Versteeg H.K. & Malalasekera W. (1995). *An introduction to Computational Fluid Dynamics. The Finite Volume Method.* England: Longman Group Ltda.

Appendix A

Poisson Pressure's Equation

The Poisson Pressure's Equation is derived from the Navier-Stokes' equation based on the prediction-correction computation. The mass conservation is written in the form of a compressible flow in order to enforce incompressibility with the deviation of fluid densities. Since, MPS aproximates the density by the particle number density,

$$\frac{1}{n_o}\frac{Dn}{Dt} + \nabla \cdot \mathbf{u} = 0 \tag{A.1}$$

From the Navier-Stokes' equation

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu \nabla^2 \mathbf{u} + \mathbf{g} \tag{A.2}$$

$$\mathbf{u}^{k+1} - \mathbf{u}^k = \left(-\frac{1}{\rho}\nabla P + \nu \nabla^2 \mathbf{u}^k + \mathbf{g} \right) \Delta t \tag{A.3}$$

the prediction step is an explicit calculation in time to obtain an intermediate velocity and position with the given viscous and the gravity terms

$$\mathbf{u}^* = \left(\mathbf{u}^k + \nu \nabla^2 \mathbf{u}^k + \mathbf{g} \right) \Delta t \tag{A.4}$$

$$\mathbf{r}^* = \mathbf{r}^k + \mathbf{u}^* \Delta t \tag{A.5}$$

The correction step is used to obtain the particles velocities and positions at the time step k+1

$$\mathbf{u}^{k+1} = \mathbf{u}^* + \mathbf{u}^{'} \tag{A.6}$$

$$\mathbf{r}^{k+1} = \mathbf{r}^* + \mathbf{u}^{k+1} \Delta t \tag{A.7}$$

where the corrective velocity is given by

$$\mathbf{u}^{'} = -\frac{\Delta t}{\rho}\nabla P^{k+1} \tag{A.8}$$

Taken the divergence of A.8

$$\nabla \cdot u^{'} = -\frac{\Delta t}{\rho}\nabla^2 P^{k+1} \tag{A.9}$$

Combinig equations A.1 and A.9 the Poisson Pressure's equation is found

$$\nabla^2 P^{k+1} = -\frac{\rho}{\Delta t^2}\frac{n_o - n^*}{n_o} \tag{A.10}$$

Appendix B

Helmholtz-Hodge decomposition

The MPS is based on the Helmholtz-Hodge decomposition of an intermediate velocity $\mathbf{u}^*$. Let $\mathbf{w}$ be a sufficient smooth vector field on a bounded doman Ω with a smooth boundary $\partial\Omega$. Then, $\mathbf{w}$ can be uniquely decomposed in the form

$$\mathbf{w} = \mathbf{v} + grad\phi \tag{B.1}$$

where $\mathbf{v}$ has zero divergence and is parallel to $\partial\Omega$ and ϕ is a scalar potential function [Chorin & Marsden (1992), Arfken & Weber (2005)]

Note that if $\mathbf{w}$ is $\mathbf{u}^*$, $\mathbf{v}$ is $\mathbf{u}^{k+1}$and ϕ is the pressure P this equation is the same as A.6

$$\mathbf{u}^* = \mathbf{u}^{k+1} - \mathbf{u}^{'} \tag{B.2}$$

or replacing $\mathbf{u}^{'}$for equation A.8 it is equivalent to

$$\mathbf{u}^* = \mathbf{u}^{k+1} + \frac{\Delta t}{\rho}\nabla P^{k+1} \tag{B.3}$$

where $\mathbf{u}^{k+1}$is required to have zero divergence.