UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and the Environment

Fluid Structure Interactions Research Group

Investigations of Smoothed Particle Hydrodynamics Method for Fluid-Rigid Body Interactions

by

Fanfan Sun

Thesis for the Degree of Doctor of Philosophy

April 2013

UNIBERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND THE ENVIRONMENT

Fluid Structure Interactions Research Group

Doctor of Philosophy

INVESTIGATIONS OF SMOOTHED PARTICLE HYDRODYNAMICS METHOD FOR FLUID-RIGID BODY INTERACTIONS

by Fanfan Sun

The aim of this project is to investigate the capability of smoothed particle hydrodynamics (SPH) method for fluid-rigid body interactions. SPH is one of the most widely used meshless methods which use particles to represent the system. The fluid is assumed either slightly compressible so weakly compressible SPH (WCSPH) is applied or truly incompressible so incompressible SPH method (ISPH) is adopted.

The performance of SPH method is affected by a number of modelling parameters including the choice of kernel functions, smoothing length, total number of particles and time step size. Investigations of the effect of these parameters were conducted using one dimensional cases and the results show that smoothing length and the total number of particles can influence the accuracy significantly but other parameters are less important.

In order to generate the model efficiently and maintain accuracy an appropriate boundary treatment is important. Two boundary treatments are investigated for ISPH method. Although these two boundary treatments have been used in WCSPH, they have not been used in ISPH method in the literature. They are easier to use for complicated engineering situations related to fluid structure interaction problems compared with the traditionally used ghost particles. Two approaches for solving Poisson's equation of ISPH method are studied including the implicit solution approach and explicit solution approach.

A new method is developed for multi-phase flow by combining WCSPH method and truly ISPH method to study the effect from air pressure. Within this method the compressibility of air and incompressibility of water can be retained.

Based on these studies, algorithms for fluid rigid-body interaction in 2 dimensional and 3 dimensional cases have been developed to simulate the general engineering problems related to fluid rigid body interactions.

Contents

List of figures

List of tables

Declaration of authorship

I, Fanfan Sun declare that this thesis entitled Investigations of Smoothed Particle Hydrodynamics Method for Fluid-Rigid Body Interactions and the work presented in it are my own and have been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;

3. Where I have consulted the published work of others, this is always clearly attributed;

4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;

5. I have acknowledged all main sources of help;

6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

7. Either none of this work has been published before submission, or parts of this work have been published as: [please list references below]:

Signed: ……

Date: ……

Acknowledgement

I would like to express my gratitude to all those people who gave me encouragement and help during my PhD life. First, of course, I would like to thank my supervisor, Dr Mingyi Tan, who introduced particle method to me and kindly explained the related fundamental numerical concepts. Especially, his help in the programming code in the early stage was very important for the later development. Professor Jing Tang Xing gave me guidance of the project and the layout of the thesis. My special acknowledgement should go to Jiapeng Liu (Tsinghua University) who helped me a lot during his visiting in our group. His knowledge of mechanics and computer language widened my understanding about this project.

Then there are graduate students in our research group, Jian Yang, Wei Wang, Keith Man, Pareecha and Qiaorui Wu working and communicating with me, their support either through vetting ideas or just learning how things work made this individual research work much more interesting and encouraged me a lot especially in the bottle neck time. Thanks to my office mates and all the colleagues in the group.

I would also like to thank Yailduo, Murat, Xiaojing, Professor Bonet (Swansea University), Dr Ben Rodgers (Manchester University) and all the people I met in the conferences for their kindness and patience of discussing all my questions.

Thanks to my landlords: Mr Anthony Robson and Mrs Ella Robson (granddad and grandma). They offered me a very cosy accommodation just like home. Most importantly, they did read my thesis and polished my written English.

Finally, a particular thank to my family, my parents and my sister. Without their support and encouragement I could not achieve so far. This work is dedicated to them.

Nomenclature

Abbreviations

ALE	Arbitrary Lagrange Euler
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CFL	Courant Friedrichs Levy
CSPM	Corrected Smoothed Particle Method
DOF	Degree of Freedom
EOS	Equation of State
FSI	Fluid Structure Interaction
FE	Finite Element
ISPH	Incompressible Smoothed Particle Hydrodynamics
MSPH	Modified Smoothed Particle Hydrodynamics
N-S	Navier-Stokes
PDE	Partial Differential Equations
PPE	Pressure Poisson Equation
RKPM	Reproducing Kernel Particle Methods
SPH	Smoothed Particle Hydrodynamics
VOF	Volume of Fluid
WCSPH	Weakly Compressible Smoothed Particle Hydrodynamics
XSPH	A variant of SPH method

Symbols

A	a generic function
a	index of a given particle
B	parameter of the equation of state
b	index of a neighbouring particle of the given particle
C	index of a cell among the total cells
c	speed of sound
D	parameter of the equation for repulsive force
dt	time step size
ds	particle spacing
E	unit matrix
e	arbitrary rotation axis
F	force
g	gravitational acceleration
H	height of water
I	moment of inertia
K	stiffness of spring
h	length scale of kernel function
M	moment
m	mass
MLS	Mean Least Squares
N	unit of force
N	number of particles
N_c	total number of cells
N_{cx}	index of a cell in x direction

N_{cy}	index of a cell in y direction
N_{cz}	index of a cell in z direction
N_m	number of time steps
$N_{max\ neighbour}$	maximum number of neighbouring particles
N_p	total number of particles
N_{pc}	maximum number of particles in a cell
N_x	number of cells in x direction
N_y	number of cells in y direction
N_z	number of cells in z direction
$\mathbf{n}$	normal direction of solid
n	number of time steps
(O)	mass centre
P	pressure
Q	Euler parameter
$\mathbf{R}$	rotation matrix
r	distance between two particles
S	surface
S	second
t	time
V	volume
$\mathbf{v}$	velocity
W	kernel function
α	artificial viscosity coefficient
β	artificial viscosity coefficient

γ	ratio of the heat capacity of the gas at constant pressure to that at constant volume
Δ	Laplace operator
ε	error
μ	dynamic Viscosity Coefficient
$\prod$	artificial Viscosity
ρ	density
σ	stress
τ	shear stress tensor
φ	a factor with small value
Ω	volume of integration
∇	Hamilton operator

Chapter 1 Introduction

Fluid structure interaction is the coupling between moveable or deformable bodies and fluid flows and the main physical characteristics is the relations between the motions and forces of both structures and fluids. Namely, fluid force acting on a structure induces deformation and movement of the structure and this in turn changes the flow and consequently the load on the solid will be further altered. The interaction of the structure with the surrounding or enclosed fluid gives rise to a rich variety of physical phenomena, for example, the response of ships or offshore structures in waves, the stability of aircraft and flutter of aircraft wings in flowing air, the flow of blood through arteries, the response of bridges and tall buildings to winds and the vibration of turbine and compressor blades. A good understanding of the dynamic interaction between the fluid and structure is very important to assess the overall performance and safety of structures in many engineering fields.

1.1 Background

1.1.1 Research of FSI in engineering fields

In many fluid structure interaction problems involving violent fluid motion, the structure may experience high stresses and encounter possible structural failure. Besides the damage of material, the motion of the solid in the fluid environment is also important for engineering design. For example, strong fluid flow may cause an aircraft to lose its stability; the coupled motion of ship and wave may cause ship to capsize. On the other hand, structure motion can induce additional fluid flow and this may further influence the normal use of the structure. Taking water spray produced by the landing gear of an aircraft running on the wet runway for example, a large amount of ingested water may cause an engine flameout. With the knowledge of the behaviour of structures in the fluid, better systems can be designed to prevent these kinds of accidents.

Research of fluid structure interaction started long time ago. A review of the history of FSI development can be found in Xing, et al. (1997) and the research involves many engineering fields:

Aircraft engineering

In aircraft engineering, strong air flow may affect the stability of the aircraft, and in some cases the structure of aircraft may be damaged under the load of the airflow. The problem associated with the fluid structure interaction involving air and deformable structures is known as aeroelasticity problem, and in fact, early research of fluid structure interaction started from this type of problem. Lanchester (1916) and Bairstow & Fage (1916) performed a set of experiments to investigate aeroelasticity vibration. The general theory of aerodynamic instability and the mechanism of flutter was established later by Theodorsen (1934). With the rapid development of aircraft industry, the aeroelasticity has become a very important research area. Much of the work has been continued for the development of the research in aeroelasticity.

Civil engineering

In civil engineering, aeroelasticity problem can also be found when the responses of bridges or tall buildings are considered subject to wind loading conditions. Apart from air flow environment, structures surrounded by water also experience the interaction with fluid. A ground-breaking research of interaction between dam and water was conducted by Westergaard (1933) and the outcome was published in his paper entitled "water pressures on dams during earthquakes". In his work the dam was assumed to be rigid and the problem is simplified as a hydrodynamic problem with known boundary conditions. In 1970, Chopra started to work on the coupling between a deformable dam and water (Chopra 1970). The design for the offshore structures under horizontal seismic loads was improved based on the researches of the coupling between the dam and water.

Sloshing of fluid in liquid storage containers is another important fluid structure interaction problem. It is an important design feature for large liquid storage containers of inflammable or explosive liquid to be used safely in earthquake condition. The fundamental theory on liquid storage container oscillation problem is reviewed by Moiseev (Moiseev 1964; Moiseev & Petrov 1964). Liquid sloshing inside the fuel tank of spacecraft may influence the stability of the craft and affect the altitude control significantly. The linear theory of small amplitude sloshing was developed first and it has been applied to the practical engineering problems. Nonlinear, large amplitude sloshing is more complicated and it is analysed mostly using the Arbitrary Lagrange-Euler method.

Marine engineering

In a violent sea, serious hydrodynamic impact may happen when there is a large vertical relative motion between the ship and wave surface. This phenomenon is called slamming, and it may cause damages to the bottom of the ship. Slamming pressures are sensitive to the way the water hits the structure and these pressures are normally neither harmonic nor periodic (Faltinsen 1993; Faltinsen, et al. 2004). Many ships have reported local structural damages due to slamming loads. For example, the tragedy of MV Estonia in the Baltic Sea on 28 September 1994, one of the deadliest marine disasters of 20^{th} century, was initiated because of the breaking of the bow door due to the severe slamming. Other situations such as large volumes of water flowing onto the deck of a ship, called green water, may cause fatal damage to the structure as well.

In the early research on ship water coupling dynamics it was assumed that a ship was rigid so that only the disturbance caused by the motion of ship was studied. Haskind (1946) constructed the velocity potential of fluid due to the motion of a rigid ship and derived the point source Green function using Green's theorem. The solution of the integral equation of velocity potential can be obtained based on the boundary conditions. This method is used for the research of the interaction between the swaying ship and water. Other research work contributing to the investigation of ship motion in waves includes: Denis & Pierson (1953) who used the spectral analysis method to calculate ship motion in irregular waves; and Korvin-Kroukovsky (1955) who used strip theory in ship motion problems (Du, et al. 2004).

The elastic deformation of a ship was studied later using hydroelasticity theory which is developed by Bishop & Price (1979). This theory has been employed to predict the responses of a wide range of marine structures. A recent review of hydroelasticity of ships can be found in Hirdaris & Temarel (2009).

Others

FSI can be found in many other engineering fields such as: blood interacting with vessels in biomechanics; metal curing process in metallurgy and casting industry; vibration of oil pipelines and the response of immersed structures to explosive waves. A good understanding of the interaction between fluid and structure is beneficial.

1.1.2 Research of FSI in academic disciplines

Fluid structure interaction (FSI) in science is an interdisciplinary subject related to fluid mechanics and solid mechanics. Fluid structure interaction problems can be classified in two major categories: in the first category, the solid and fluid are well mixed without a clear interface between the two phases and one example is the saturated soil; in the second category, there is a clear interface and the interaction happens only on the interface. In the second category there are further three different cases: the first case is the aero-elasticity problem involving structure vibration in air flow; the second one is the slamming problem involving finite movement of fluid in a short duration and the third case is conventional ship motion problems involving oscillatory fluid motion in a long period of time. For all these types of FSI problems, the solution procedure has to be based on the understanding of the relationship between the forces.

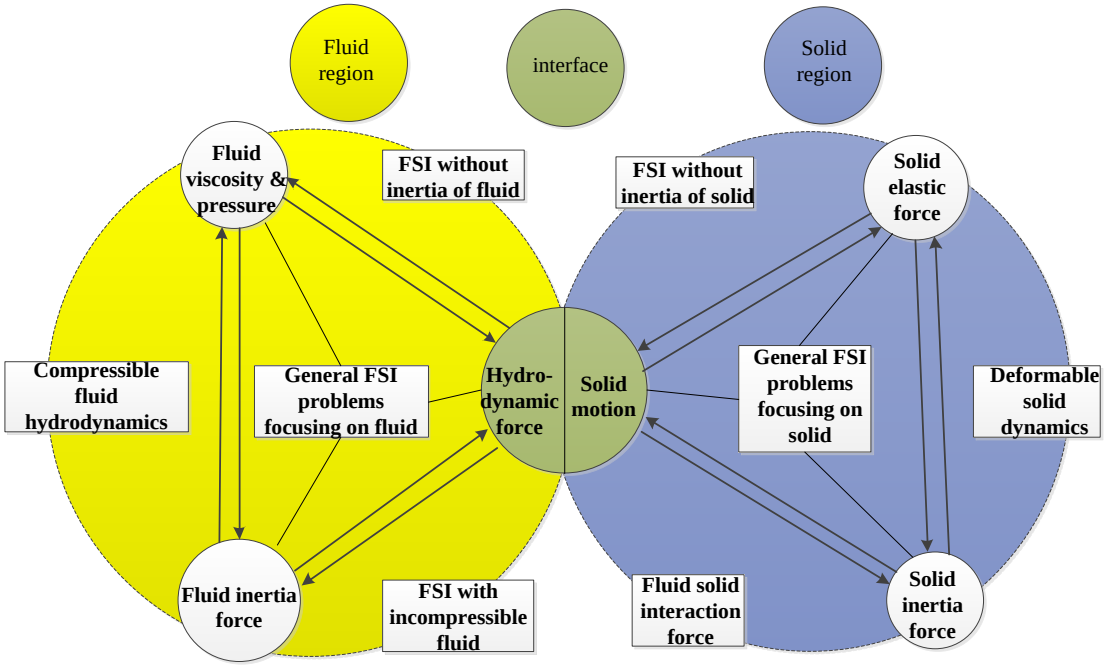


Figure 1.1: Overall view of fluid structure interaction (Xing, et al. 1997)

The relationship among different types of forces in fluid solid interaction is shown in Figure 1.1. The two large circles represent fluid and solid regions, respectively, and the smaller circle in the middle represents the interface. A fluid hydrodynamic force influences the motion of a solid through the interface and this in turn affects the fluid. The hydrodynamic force and the motion of the solid are all unknown on the interface

and they can only be solved based on the physical descriptions of the whole system. In this case, it is important to consider the interactions. If the motion of the solid is given or the hydrodynamic force is known then the problem will become a fluid hydrodynamic problem under known boundary conditions or a solid dynamic problem under given wet interface traction.

Fluid structure interaction is closely related to a few scientific research areas such as fluid hydrodynamics, solid dynamics, nonlinear mechanics, and numerical methods, etc. Progress in FSI research will have an impact on these related academic areas and this can result in new improvement. Hence, FSI research is significant for science and engineering development.

1.2 Solution approaches for fluid structure interaction

In order to solve fluid structure interaction problems it is necessary to determine variables related to fluids and solids at the same time. These variables normally cannot be determined individually. However, an FSI problem can be simplified for specific research purposes. For example, the compressibility of water can be ignored in long term water structure interaction problems without the need to consider acoustics and in some cases the deformation of the solid can be ignored so that the solid is treated as a rigid body.

For almost of the fluid structure interaction problems there is no analytical solution because of the complexity of the problem, so numerical solutions or experimental studies are the only way forward (Xing, et al. 2003). Experiments are normally expensive to perform so numerical methods are preferred in many cases. Currently, finite element (FE) method and computational fluid dynamics (CFD) are the mature numerical methods for structure and fluid analysis respectively.

In FE method, the continuous problem domain is divided into a number of discrete subregions or “elements”, connected at discrete points called “nodes”. The solution for the whole domain is represented by a collection of the solution on each element. The value of a generic function at one element can be derived using the known values at the nodes of this element according to the interpolation function. Thus, the original problem with infinite degrees of freedom is replaced by a problem with finite degrees of freedom and a function on a continuous field is approximately represented by a

collection of functions of each elements. In CFD, the domain is discretized as a finite set of control volumes or grids. General conservation equations for mass, momentum, energy, etc., are discretized as algebraic equations. In the discrete domain, each flow variable is defined only at the grid points. The values at other locations are determined by interpolating the values at the grid points. The set of equations are solved simultaneously to determine flow field.

FE method and CFD method are based on different descriptions. FE method for structure is usually described using Lagrangian formulations, whereas CFD for fluid is described using Eulerian formulations (Bathe & Zhang 2004). In the Lagrangian description, the grid or mesh is fixed on the material and it deforms with the material. The physical properties of a particular point of the material at a time instant are solved and recorded (Price 2006). In the Eulerian description, the grid or mesh is fixed in the space and the material moves across the grid so the physical quantities at a point fixed in space are recorded.

In fluid structure interaction problems, when a structure element moves, the material coordinates will move with the element to new positions in space while the Eulerian coordinates that describe the fluid remain unchanged. The difference of these two descriptions creates a separation of the mesh points between solid and fluid.

1.2.1 Linear problems

For small-disturbance problems, this separation can be neglected and the mathematical model is formed based on the original static equilibrium configuration of the fluid-solid interaction system on which a numerical analysis is developed. For these linearized problems, the superposition principle and the mode theory of structural analysis are applicable (Xing, et al. 2003). Mathematical equations and associated solution procedures for these problems are well developed in literature (Bishop & Price 1979; Bishop, et al. 1985; Xing & Price 1991; Xing, et al. 1996).

1.2.2 Nonlinear problems

For nonlinear problems Navier-Stokes (N-S) equations or Euler equations are used for fluid. Here, two different cases can be considered: i.e. a weak interaction case and a strong coupling case depending on whether there is a small or large deformation of the fluid and structural domain (Rugonyi & Bathe 2001). Especially for strong coupling cases, the difference between Lagrangian and Eulerian descriptions has to be

fully accounted for to ensure the validity of compatibility conditions on the coupling interfaces (Xing, et al. 2003).

1.2.3 Solution with Arbitrary Lagrangian-Eulerian method

It is difficult to enforce the kinematic compatibility on the fluid structure interface if there is a large structural displacement. To overcome this problem, an Arbitrary Lagrangian-Eulerian (ALE) numerical model was developed using finite element methods in both the solid and fluid domains (Donea, et al. 1977; Belytschko, et al. 1982; Bathe, et al. 1995). Alternatively, Xing, et al. (2003) adopted the ALE finite difference technique proposed by Hirt, et al. (1974) to calculate the fluid at all speeds based on a moving coordinate system fixed in the structure. This coordinate system is used to describe fluid flow and to construct structure-deformation equations.

In the ALE description, the nodes of the computational mesh may move with the material in Lagrangian fashion, or they can be fixed in Eulerian manner. When ALE technique is used in engineering simulations, the computational mesh inside the domains can move arbitrarily to optimize the shapes of elements, while the mesh on the boundaries and interface can move with the materials to track the boundaries and interfaces of a multi-material system precisely. Because of this freedom in moving the computational mesh offered by the ALE description, greater distortions of the material can be handled than would be allowed in a purely Lagrangian method and this is with better resolution than a purely Eulerian approach. For fluid structure interaction, fluid flow is described using Eulerian description so that flow calculations can be carried out on continuously deforming meshes while the solid motion is described using Lagrangian description and the mesh representing solid is glued to the material (Khurram & Masud 2006).

However, fluid elements tend to become distorted in the case of large solid translations and rotations. And the accuracy of solution would deteriorate due to increased anisotropy or uneven distribution of the grid points. In this case, remeshing is often required but it can be quite time consuming (Loon, et al. 2007). Besides, it is difficult to use as it requires a transformation between these two different descriptions (Carlton 2004).

To summarize, these traditional numerical methods require the continuum to be divided into linked small elements or volumes as shown in Figure 1.2. These elements, called mesh or grid, allow the governing partial differential equations to be converted

into a set of algebraic equations. The governing equations are formulated in either Lagrangian or Eulerian description. Grid-based numerical methods experience difficulty in solving certain type of problems typically associated with large deformations. Especially, the grid generation is not always straightforward when dealing with complex problems and the mesh or grid will be distorted when the boundary deforms and this may result in a break-down of the computation (Jenssen, et al. 1998). Therefore, an alternative numerical method without grid or mesh is required to overcome these problems due to mesh dependency.

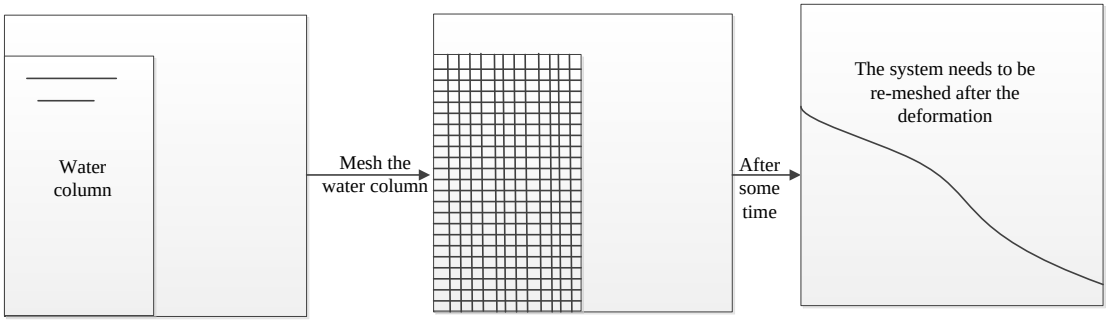


Figure 1.2: An example of water column mesh

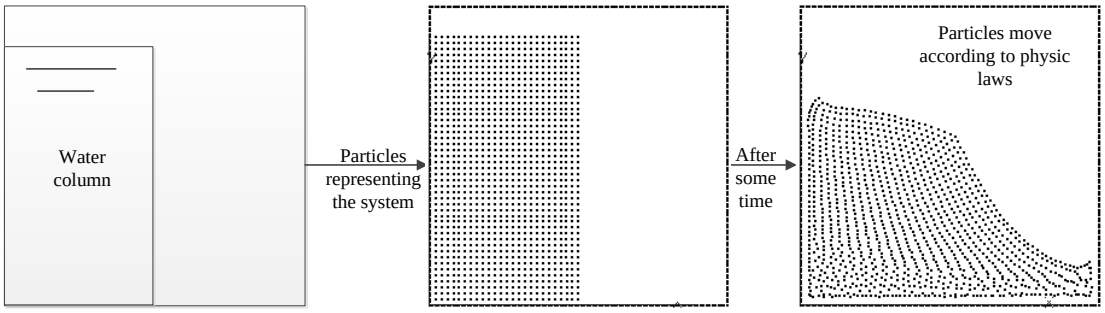


Figure 1.3: An example of the system represented by particles

1.2.4 Solution with meshless method

In a meshless method, the system is represented by a set of discrete particles as shown in Figure 1.3. Each particle carries physical properties such as mass, momentum, energy, etc. The movement of the particles is governed by associated conservation laws.

Smoothed Particle Hydrodynamics (SPH) method is one of the earliest and most widely used meshfree methods. It has been combined with FE method to be applied to FSI problems in the literature. The problem with this hybrid method is that it is hard to guarantee an accurate information transmission between particles and elements on

the interface. Alternatively, using SPH for both fluid and structure will keep the system consistent so the algorithm is simplified. However, using particles to represent the whole system requires large memory and long computational time. Besides, accuracy of the results is difficult to guarantee because of the natural drawbacks of particle method. Since the evolution of the field depends on the distribution of particles, the results is sensitive to particle distribution. However, it is difficult to guarantee a good particle distribution once they started to move following the related physical laws. In addition, some of the coefficients used in the particle formulations are selected based on experience and their influence on the accuracy is not entirely clear. Further research is necessary to improve the performance of SPH method on FSI problem.

1.3 Objective

The aim of this project is to improve SPH method for fluid structure interaction problems involving a truly incompressible fluid and rigid body. First, the potential factors influencing the accuracy of SPH approximation need to be investigated to understand how to control or improve the performance of this method. This will be carried out by using SPH approximation for several 1D functions. Different from other research work that has been done theoretically to analysis the accuracy, consistency and stability of SPH method (Liu & Liu 2003b), these properties will be investigated through numerical experiments using various kernel functions with different smoothing lengths and numbers of particles so that the influence of these factors can be observed directly.

Second, methods to ensure incompressibility of the fluid will be studied in terms of accuracy, stability and CPU time requirement through numerical experiments and the preferred method will be selected. And then boundary treatment needs to be studied taking into consideration of the trade-off between the accuracy and efficiency for the selected method. Solution approaches for the algorithm of the selected method needs to be considered and studied to ensure an efficient simulation in terms of CPU time requirement and accuracy. In addition, neighbouring particle searching strategy needs to be formulated to reduce the computational cost.

Third, air water two-phase flow will be considered to study the effect of air. A new approach will be developed for this case regarding that the compressibility of these two phases is different and large density ratio is involved.

Finally, an algorithm of 3D fluid rigid body interaction simulation needs to be accomplished based on the investigations mentioned above. And this algorithm will be applied to a few examples in both 2D and 3D to demonstrate the performance.

1.4 Layout of the thesis

The layout of the thesis is illustrated in Figure 1.4 below.

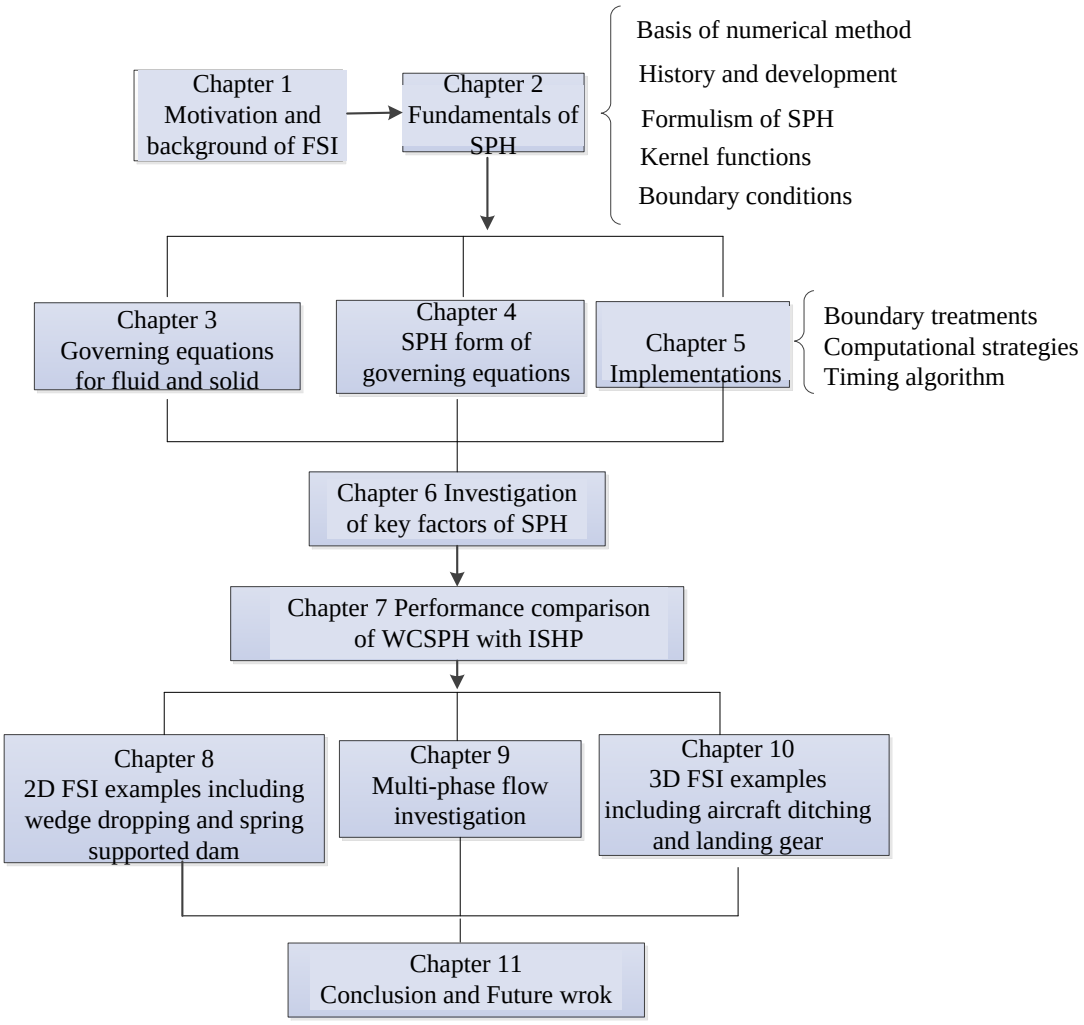


Figure 1.4: Layout of thesis

The background of fluid structure interaction and numerical method has been introduced. In the following chapter the fundamentals of smoothed particle hydrodynamics method will be discussed. Governing equations for fluids and solids will be given in Chapter 3. Numerical algorithm is described in Chapter 4. The implementation including boundary treatments, computational strategies and time stepping algorithm will be discussed in Chapter 5. The studies of the effect of modelling parameters in SPH such as kernel function and number of particles will be shown in Chapter 6. The performance of the two main SPH methods is compared using dam breaking case study in Chapter 7. Applications to 2D fluid rigid body interaction and multi-phase flow will be given in Chapters 8 and 9 respectively. 3D examples are shown in Chapter 10. Finally, the conclusion and future work is given in Chapter 11.

Chapter 2 Literature review

2.1 Meshless method

The early development of meshless methods can be traced back to 1970s. The main aim is to develop accurate and stable numerical solution procedures using a set of arbitrarily distributed particles without imposing any connectivity condition on these particles. The initial work in developing meshless methods is mainly to modify the conventional grid required methods in order to make them more adaptive and robust. The applications of meshless methods are concentrated on problems to which the conventional grid based methods are difficult to apply, such as problems with free surface, deformable boundary, large deformation that requires complex mesh generation and adaptive mesh requirement, etc.

A number of meshfree methods have been proposed to analyse solids and fluids. Most of the meshless methods are inherently Lagrangian methods. According to the numerical discretization technique used, meshfree methods can be classified in three types: 1) methods based on strong form of formulations, 2) methods based on weak form of formulations, and 3) particle based methods. A strong form of system equation expressed in ordinary differential equations (ODE) or partial differential equations (PDE) is derived based on the theory of continuum mechanics. The strong form methods normally use the collocation approach and the system is represented by collocation points (Liu & Gu 2005). They are computationally efficient but they are often unstable for irregularly distributed nodes. For a weak form an integral operation is applied to generate the discrete system equations. Formulation based on weak forms can usually produce stable and accurate results. However, in the weak form method a background mesh is required for the integration of the weak forms so it is not entirely mesh free. Details of many existing meshfree methods can be found in monographs by Liu (Liu 2002; Liu & Liu 2003b).

In a particle based method the system equation is in the strong form but its implementation is very similar to the weak form method and no background mesh is needed. The integral operation is applied in the stage when function is approximated rather than in the stage when the discrete system equations are generated as in the

normal weak form method. Smoothed Particle Hydrodynamics (SPH) method is one of the earliest and most widely used meshfree particle based methods.

2.2 History and development of smoothed particle hydrodynamics method

Smoothed Particle Hydrodynamics (SPH) has been developed largely in the last three decades since its introduction in Astrophysics in 1970s (Gingold & Monaghan 1977; Lucy 1977). It is based on the theory of integral interpolant (Monaghan 2001) and the partial differential equations are approximated by integral formulation involving a kernel function. A kernel function should satisfy a few conditions such as that it should behave like a delta function, with compact support and be integrated to unity. More discussion on this will be given in section 2.4. The interpolation method used in the particle method is closely related to the standard interpolation methods used in other more traditional numerical methods such as finite element method (Monaghan 1982). Apart from interpolation methods, SPH formulae for governing equations can be derived based on Lagrangian formulation as shown by Bonet, et al. (2004).

In the SPH method, the system is discreticized into many particles which carry material properties such as density, velocity, stress and so on. The integral representation of the function is approximated by summing up the contribution from the nearest neighbour particles defined by the kernel function. A number of particle approximation forms of a function can be derived based on different mathematic manipulations and some of the details will be shown in Chapter 3.

Unlike traditional mesh required methods, in a particle based method, the particles move following the physical laws without explicit connections between each other. Therefore, this method is most suitable for the simulation of fragmented fluid such as sprays, breaking waves and explosions, flow with free surface and other large deformation problems.

2.2.1 Application to incompressible fluid

In the early applications of SPH to incompressible fluid problems (Monaghan 1994), the fluid such as water was assumed to be slightly compressible. Therefore, quasi-incompressible equation of state was used to calculate pressure. The time step size depends on sound speed which was adjusted to restrict the fluid density variation.

This approach that treats the incompressible flow as a slightly compressible flow is named as weakly compressible SPH (WCSPH) method and it is proved to be able to successfully simulate Poiseuille flow as the result generated agrees well with the data from the finite volume method (Lobovský & Vimmr 2007).

However, WCSPH requires a very small time step and even small density error can cause significant unphysical pressure fluctuation (Lee, et al. 2008). In order to overcome these problems, an approximated pressure projection method was developed by Cummins & Rudman (1999). The incompressibility is enforced by solving a pressure Poisson's equation. Afterwards, a truly Incompressible SPH method (ISPH) was proposed by Shao & Lo (2003) where prediction-correction fractional time steps are used to update the related physical properties. In this method, the intermediate velocity field is integrated forward in time without considering the pressure effect in the first step. The temporal density obtained from the first step is then implicitly projected onto a velocity divergence-free space to satisfy the incompressibility requirement in the second step. The pressure values are calculated through a Poisson's equation. This truly incompressible SPH has been used since then (Hosseini, et al. 2007).

Alternatively, the incompressibility can be enforced by setting the volume of each fluid particle as a constant using Lagrangian multiplier in the simulation (Ellero, et al. 2007). However, a more straightforward way to ensure the incompressibility is to use constant density thus only the velocity divergence free condition is considered according to the continuity equation (Lee, et al. 2010).

The major difference between WCSPH and ISPH methods are about the pressure calculations. Normally, WCSPH is fully explicit and ISPH requires an iterative solution approach for the Poisson's equation. It was found that pressure values obtained from WCSPH method are not accurate and fluctuate severely because they largely rely on the changes of the density, any small density change will lead to a large pressure oscillation (Lee, et al. 2010; Antuono, et al. 2012), this will be shown in Chapter 7. However, many research works have been focusing on improving the accuracy of density estimation and eliminating the fluctuation of pressure for WCSPH as it is easier to parallelize (Lee, et al. 2010; Antuono, et al. 2012) and the free surface condition is implicitly satisfied (Colagrossi, et al. 2009). Colagrossi & Landrini (2003) suggested to filter the density field through a Mean Least Square (MLS) integral interpolation. Alternative, diffusive terms are added in the continuity equation to

reduce the numerical noise inside the density field (Ferrari, et al. 2009; Molteni & Colagrossi 2009; Antuono, et al. 2010) or correction terms can be used to adjust the particle displacement to ensure uniform particle distribution (Ozbulut, et al. 2012; Shadloo, et al. 2012). In contrast, ISPH method can be more accurate and produce pressure fields effectively. However, particle distributions may become highly distorted as a result of simulation errors and consequently instability arises. Xu, et al. (2009) suggested a new stabilisation technique by shifting the particles slightly to avoid the instabilities due to particle stretching. Using this shifting algorithm can improve the results but more computation is required especially when large number of particles is used. In this project, the shifting algorithm is not used considering the computation expenses; the performance of WCSPH and ISPH is compared including the accuracy of results, CPU time requirement and stability of the algorithm, so that one of them can be selected for the investigation of fluid rigid body interaction.

2.2.2 Application to multi-phase flow

Multi-phase flow is common in nature. Neglecting the effect of one phase in flows may result in incorrect approximation. Multi-phase flow needs to be considered to investigate the influence of the entrapped fluid.

The early SPH application to multi-phase flow is for compressible fluids such as dusty gas. The mixed fluid is treated as a new type of fluid. The mass density of this new fluid can be updated based on the continuity equation and the pressure can be calculated based on the equation of state. A void fraction is used to account for the contributions of each individual fluid to the mixture. Since the densities of the dust and dusty gas are known, the density of gas can be obtained through the void fraction. With the known pressure and density values, the velocity of each phase can be determined based on the momentum equation. Hence, the motion of the whole fluid can be determined (Monaghan & Kocharyan 1995; Johnson & Beissel 1996a).

It is more difficult to consider the situation when air is mixed in water. A large density ratio may cause instability for the algorithm since the conventional particle formulation is based on the assumption of a continuous material or small density variation. In a method developed by Ritchie & Thomas (2001) the pressure of gas is assumed to be constant as the sound-crossing time is shorter than the flow time across the smoothing sphere. The density was calculated from the equation of state so that the discontinuity of density would not affect the density update in the simulation.

Alternatively, Colagrossi & Landrini (2003) derived a modified particle evolution form to avoid differentiation through the interface where density discontinuity occurs. However, the conservation of mass is not satisfied with this evolution equation so that normally a density re-initialization approach is needed (Monaghan 1992). Another particle evolution form was derived by Hu & Adams (2006) using a Shepherd function to represent the particle volume which is usually represented by the relationship between mass and density. By doing so, the density term is removed from the formulas thus the large density difference is not a problem any longer. The derived particle evolution form is symmetric so that the momentum is conserved. Similar forms of particle evolution are derived based on different mathematic considerations (Hu & Adams 2006; Grenier & Touze 2008; Grenier, et al. 2009). Hu & Adams (2007) developed an incompressible multi-phase SPH method by applying the modified particle evolution form to ISPH algorithm. In this method the pressure of the fluids is calculated by solving Poisson's equation and the density is updated according to the continuity equation.

Both WCSPH and ISPH methods can be applied to multi-phase flow successfully. Another consideration for incompressible fluid is that the density can be assumed to be constant. In this case, none of the existing multi-phase methods is applicable to the problem when liquid is mixed with gas. Therefore, a new method can be developed to combine these two methods, namely, to use WCSPH method for compressible fluid phase and ISPH method for incompressible fluid phase. Details of this new method will be included in the following chapters.

2.2.3 Application to FSI problems and solid

Many numerical methods have been developed to analyse the fluid structure interaction problems. Due to large motions normally found in fluids, a meshless method is a useful option for flow simulations. Finite element method can be used reliably for structure analysis. Therefore, it is viable to use meshless methods for fluids and finite element method for solids in a fluid structure interaction analysis.

One of the first coupling procedures for meshless particles and finite elements was proposed by Attawy, et al. (1994) and Johnson (1994). They adopted a commonly used coupling algorithm called: master-slave algorithm (Belytschko, et al. 2000), to couple the fluid structure interactions (Johnson & Beissel 1996a). The contact constraint was imposed by applying a contact force to both the slave node (particles of

fluid at the interface) and the master surface (finite element at the interface). This is used to prevent particle penetration in the time stepping procedure. Here, the force is normal to the corresponding element surface and sliding between particles and elements in tangential direction is allowed. In the algorithm, if there is a movement of the slave node, the master nodes will move in a manner consistent with the velocity changes. When a slave node overlaps the master segment, the normal velocities of these three nodes involved are artificially adjusted to conserve linear momentum and angular momentum. Details on coupling of meshfree methods and finite elements can be found in Rabzuk, et al. (2000).

The FEM and SPH combination is capable of simulating fluid structure interaction successfully. However, the implementation on the interface is complicated because either the data need to be transferred between two different methods or a contact algorithm is required. The simpler approach is to apply the SPH method to the analysis of solid as well. The shear stress and pressure formulae can be derived by applying SPH method directly to the strain rate tensor. This makes the transfer of information between the fluid and structure domains easier as a similar method is used for both parts. It also makes the simulation more efficient in the case where large deformation happens in the solid.

There are two coupling models when SPH is used for both solid and fluid: one is to treat all the particles in the same way regardless of their nature and an XSPH correction is applied to stop the particle penetration (Rafiee & Thiagarajan 2008; Rafiee & Thiagarajan 2009) whereas the other is to determine the exact position of the interface and its normal direction before the force and the reaction are calculated (Antoci, et al. 2007). In most cases artificial viscosity and artificial stress can be used to improve the stability (Antoci, et al. 2007; Bui, et al. 2007). In this project, the solid is assumed to be rigid and the whole system of fluid-rigid body interaction is represented by particles. The second treatment on the interphase is used as each phase is computed in separated algorithms. This will be detailed in Chapter 3.

When SPH is applied to the solid in a state of isotropic tension, the solid is stretched and the SPH particles attract each other to resist the stretching. This can cause clumping of the particle and hence the use of the standard SPH equation in conjunction with explicit time stepping scheme can lead to unstable time integration for any time step size. This type of instability is called “tensile instability” and it is reported that it cannot be eliminated by introducing artificial viscosity (Bonet & Lok

1999). This is a major deficiency of traditional SPH. A number of methods were developed to improve the accuracy, stability and consistency of SPH.

2.2.4 Variations of SPH methods

As SPH has some drawbacks such as tensile instability for solid simulation and inconsistency near a boundary, various methods have been developed for different specific purposes to improve the original SPH method. Liu, et al. (1995b) developed a reproducing kernel particle methods (RKPM) to improve the accuracy of SPH approximation especially around the boundaries. This method composed of a correction function and a window function (same as the kernel function) which was originally proposed in the theory of wavelets (Chui 1992). The correction function can be expressed as a linear combination of polynomial basis function with unknown coefficients determined to ensure the approximated function or the derivative of the approximated function to be reproduced exactly. The number of these coefficients involved in the definition of the correction function depends on the order of the highest derivative term presented in the governing equations (Aluru 1999). This method eliminated the tensile instability associated with SPH methods (Liu, et al. 1995b; Jun, et al. 1998).

Another method proposed to reduce the tensile instability as well as boundary effects uses the normalized smoothing function which is adjusted for every particle to normalize the kernels (Johnson & Beissel 1996a). This algorithm was applied to cylinder impact problems (Johnson, et al. 1996b; Johnson, et al. 1996c).

For the problems of unsteady boundary values such as heat conduction, a corrective smoothed particle method (CSPM) was developed by Chen, et al. (1999a). The SPH evolution formulations are derived based on the Taylor series of the function. The number of the terms involved in the Taylor expression depends on the order of the approximated function. This method is further applied to nonlinear dynamic problems including transient heat conduction and structure dynamics (Chen & Beraun 2000).

By modifying CSPM, a modified smoothed particle hydrodynamics (MSPH) method was developed (Zhang & Batra 2004) to improve the accuracy of the approximation near the boundary but it is more time consuming since a more complicated solution process is required. This method is further formulated in

cylindrical coordinates to analyse axisymmetric deformations of a circular cylindrical body (Batra & Zhang 2008).

To adjust the resolution of the particle simulation, Lastiwka, et al. (2005) proposed an adaptive particle distribution method for SPH. Any number of particles can be removed or inserted and this was found to improve the accuracy in a shock tube simulation.

With those modifications to the formulations the performance of SPH was improved for specific applications accordingly. However, the implementation is more complicated to a certain degree. Hence, research has been focused on how to select the modelling parameters which may influence the performance of the original SPH method such as kernel function, smoothing length, and particle distribution. A brief understanding of the influence of different choices of kernel functions, number of particles and smoothing lengths are given in Chapter 6.

2.2.6 Accuracy of implementation

From the analysis of the truncation error of the gradient approximation carried out by Quinlan, et al. (2006) in a study of the robustness and accuracy of SPH formulations, it is concluded that 1) a uniform distribution of particles is better for obtaining accurate results; 2) a smaller smoothing length gives more accurate results provided that there are sufficient neighbouring particles in the smoothing domain and 3) for non-uniformly distributed particles the accuracy of SPH discretization can be improved if the absolute values of pressure and velocity is reduced before calculating the gradients. In the example of Poiseuille flow, by subtracting hydrostatic pressure from the absolute pressure in the momentum equation, the (absolute) value of pressure was reduced so was truncation error (Basa, et al. 2009).

In some cases, particle oscillation may happen and this can result in incorrect approximation. In this case, the original SPH is not able to provide accurate estimations therefore correction is necessary. Usually, an artificial viscosity term is added to the momentum equation to eliminate the instability (Monaghan 1994). Furthermore, in an XSPH method, the velocity is modified by artificially adding an averaging term from the neighbouring particles (Monaghan 1989; 1992; 2002). Namely, the XSPH method adjusts the velocity of a particle so each particle moves with a velocity closer to the average velocity of the neighbouring particles. This velocity corrective term is used to smooth out oscillations of particle velocities

calculated by integration of the momentum equation (Antoci, et al. 2007; Crespo, et al. 2007; Lobovský & Vimmr 2007).

Other aspects of the numerical implementation of SPH including boundary treatments, construction of artificial viscosity and nearest particle searching algorithm can be found in the early works (Monaghan 1988; Monaghan & Lattanzio 1985b; Monaghan 1992). They will be explained in detail in Chapter 5.

2.3 Kernel function

The most important aspects of SPH method are the choice of kernel function and boundary treatment. The kernel choice has to be constructed first for a successful SPH simulation. An improper choice of the smoothing function may lead to unphysical structures of the system (Schussler & Schmitt 1981). A kernel function $W(\mathbf{x},h)$ is usually an even function of **x** because of this the error terms of the kernel integral, when expressed in Taylor expansion, involving odd powers of **x** will vanish. This is equivalent to liner interpolation (Monaghan 1982).

A kernel must satisfy several conditions in order for the SPH model to satisfy the requirement of interpolation theory (Monaghan 1992)

The first condition is the normalization condition

$$\int_{\Omega} W(x,h)dx = 1 \qquad (2.4.1)$$

The second is the Delta function condition. Namely, when the smoothing length approaches to zero the kernel should approach to the Delta function:

$$\lim_{h\to 0} W(x,h) = \delta(x) \qquad (2.4.2)$$

And the third is the compact support condition

$$W(x,h) = 0, |x| \geq kh \qquad (2.4.3)$$

where *k* is a constant which defines the support domain (i.e. smoothing length) of the smoothing function as shown in Figure 2.1. The normalization condition ensures that a continuum function can be approximated to the zero-th order. With an even function condition and the normalization condition, the kernel approximation will have second order accuracy $O(h^2)$. This can be proved by using Taylor series expansion for the SPH integral representation. The Delta function condition makes sure that the approximation value approaches the function value as the smoothing length tends to zero. The compact support condition transforms a SPH approximation from global operation to a local operation.

In addition, a kernel function must be positive within the smoothing domain. This is not necessary mathematically but it is important for physically. A negative kernel function may lead to unphysical parameters such as negative density and energy. A kernel aiming to produce better approximation should have smoother values of the function and its derivatives. This is because a smoothing function will not be sensitive to particle distribution (Liu & Liu 2003b). From Equation (2.3.8), it is clear that at least the first derivative of the kernel function should be continuous so that derivatives of the function can be approximated. And if the second derivative is continuous, the kernel is not sensitive to particle distribution (Monaghan & Lattanzio 1985b)

It should be possible to use different kernels for different property calculations. Fulk & Quinn (1996) proposed a measure of merit for SPH kernels in the form of second order integral approximation based on an analysis of kernels in 1-D case. In this work it was found that the shape of the kernel function and the distance ratio between particle spacing and smoothing length are the two key factors influencing the kernel approximation. Kernel functions with bell shape, hyperbolic shape, parabolic shape and double hump shape were considered. It concluded that the bell shaped kernel functions outperform other kernels and a smaller distance ratio is better for more accurate results. For a give function such as spline functions which satisfies the three conditions for a kernel function as given in Equations (2.4.1) to (2.4.3), the criteria for the kernel or parameter choices is to reduce the difference between the following two functions (Capuzzo-Dolcetta & Lisio 2000) :

$$\langle \nabla f \rangle_1 = \int f(\mathbf{x}') \cdot \nabla W(\mathbf{x}' - \mathbf{x}, h) d\mathbf{x}'$$

and

$$\langle \nabla f \rangle_2 = \nabla \int f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' .$$

A general kernel constructing method based on the consideration of restoring particle consistency is proposed by Liu, et al. (2003a).

Smoothing domain

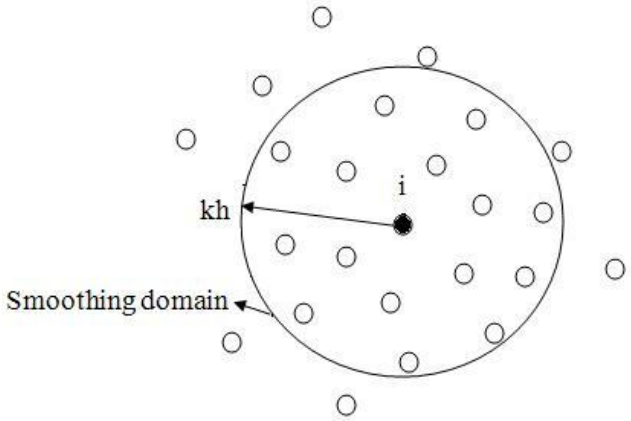


Figure 2.1: Smoothing domain of a particle

For a specific particle, only the particles inside its smoothing domain contribute to the calculations. These particles are called neighbouring particles. An appropriate smoothing domain is important to ensure a correct SPH approximation. Normally, it should be large enough for each particle to have enough neighbouring particles but is should be small enough to preserve the accuracy as well as to reduce computational cost. The spacing between particles is also important. If the particle spacing is too large, the accuracy of the approximation will be affected because of the averaging effect whereas if it is too small the computational cost will increase. Therefore, an appropriate particle spacing should be selected for a specific problem to ensure the best performance of the simulation. The ratio between particle spacing and the radius of the smoothing domain is normally taken as 2.5 and this will be shown in Chapter 6 based on the investigation of accuracy using different number of particles and smoothing lengths.

Chapter 3 Fluid-rigid body interaction problems

3.1 General description

In fluid structure interaction problems, the solid can be rigid or deformable. In the current project the point of interest is the motion rather than the material behaviour, the deformation of the structure is not considered so that the solid is assumed to be a rigid body.

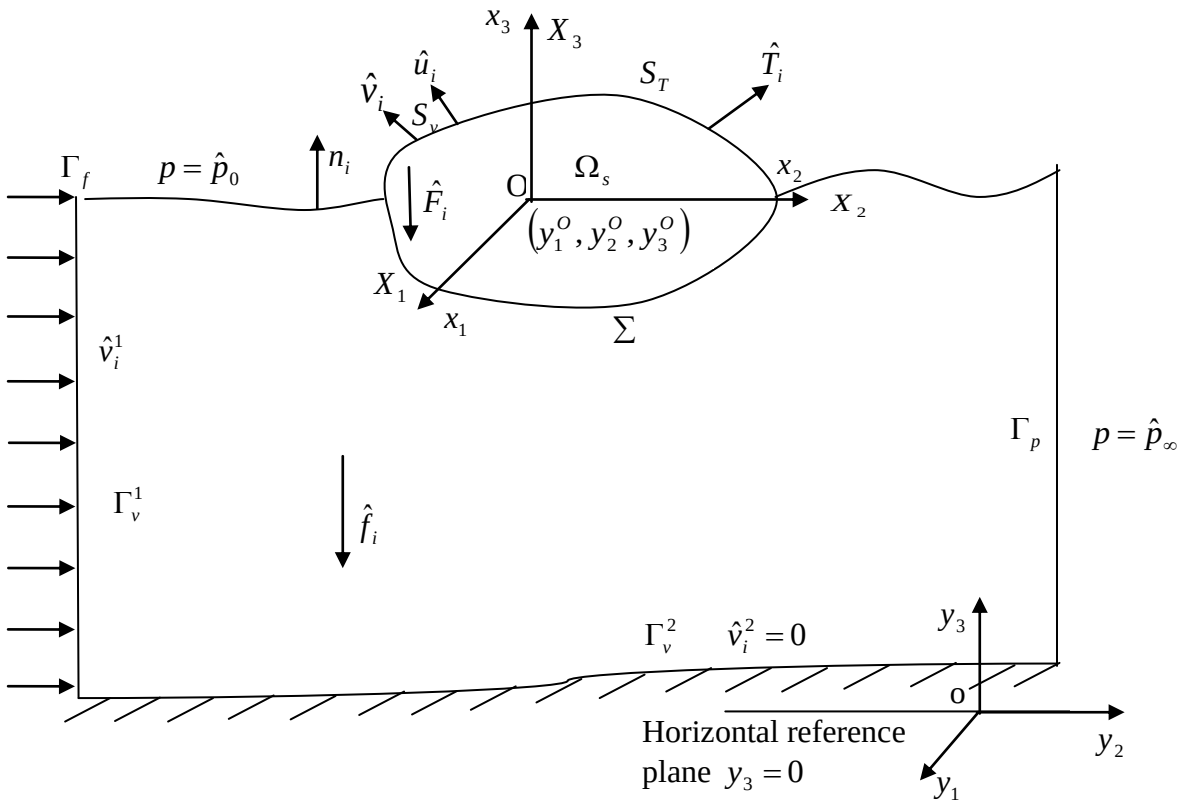


Figure 3.1: A rigid body floating on free surface

Figure 3.1 illustrates an arbitrary rigid body floating on the surface of water (Xing, et al. 2003). The water can be calm or with violent motion. The rigid body may float freely on the surface with given initial velocity or it may have a forced motion in the water. In this figure, $oy_1y_2y_3$ represents a global Cartesian coordinate system; $Ox_1x_2x_3$ denotes a moving coordinate system parallel to the global coordinate system but with its origin located at the mass centre of the rigid body; $OX_1X_2X_3$ is the body-

fixed coordinate system fixed on the rigid body and it is assumed to coincide with the moving coordinate system at the beginning. With a disturbance, the mass centre (O) is allowed to move transnationally with acceleration w_i^o, velocity v_i^o and displacement u_i^o and the body is allowed to rotate with angular displacement θ_i around the mass centre. The moving system $Ox_1x_2x_3$ is used to describe the translation of the rigid body which can be represented by the motion of the mass centre (O). For each point on the rigid body, the relationship between the coordinates in the moving system and those in the global system is given as:

$$y_i = y_i^o + x_i, \quad \dot{y}_i = v_i^o + v_i \tag{3.1.1}$$

where y_i^o represents the coordinates of the mass centre in system $oy_1y_2y_3$ and $\dot{y}_i$ and v_i are the velocities at a point on the rigid body relative to the global and moving systems respectively.

On the surface of solid domain Ω_s, one part is the interface $\sum$ with fluid, other part S_T is subject to known external traction forces $\hat{T}_i$ and given displacement u_i (which is ignored for rigid body case). On the free surface Γ_f of fluid, pressure is known as the atmospheric pressure $\hat{p}_0$ acting in i direction which is perpendicular to the free surface. The flow velocity $\hat{v}_i^1$ is given on boundary Γ_v^1 whereas fluid pressure $\hat{p}_\infty$ is known on boundary Γ_p.

With the known forces and boundary conditions the motion of the fluid-rigid body interaction system can be determined. The description of the motion of a rigid body will be introduced first and then the governing equations for the motion of solid and fluid will be formulated.

3.2 Motion of a particle in the solid

Since there is no deformation on a rigid body, for each particle in the body, its position is determined by the translational motion of the body at its centroid of mass and the rotational motion about the centroid of mass. Namely, the motion of a particle

X_i in the rigid body can be calculated based on the translational displacement u_i^o, velocity v_i^o and acceleration w_i^o of its mass centre and its rotation about the mass centre using the following formulations

$$u_i = u_i^o + \tilde{u}_i \tag{3.2.1}$$

$$v_i = v_i^o + \tilde{v}_i \tag{3.2.2}$$

$$w_i = w_i^o + \tilde{w}_i \tag{3.2.3}$$

Here, "~"denotes the variables describing the particle motions relative the mass centre due to rotation. The translation of the mass centre (O) is given in the global coordinate system $oy_1y_2y_3$ and the relative motion of a particle is described by a moving coordinate system $Ox_1x_2x_3$. Hence, translation is straightforward to calculate, while the relative motion (rotation) needs to be transformed from a body-fixed coordinate system $OX_1X_2X_3$ to a moving coordinate system and a coordinate transformation matrix is then required.

Coordinate transformation matrix

Assuming that the body-fixed coordinate system $OX_1X_2X_3$ coincides with the moving coordinate system $Ox_1x_2x_3$ initially, after a given rotation, the orientation of the axes will be different as shown in Figure 3.2.

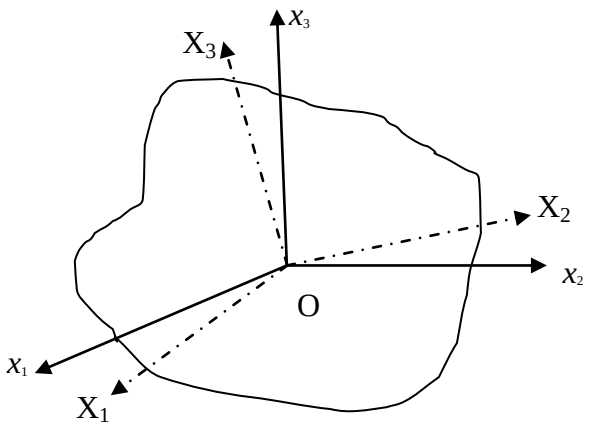


Figure 3.2: Relative rotation between body-fixed coordinate and moving coordinate systems

The new position of the rigid body in the moving system can be defined by this relative rotation. The relative rotation between the two coordinate systems can be expressed by the direction cosine matrix $\mathbf{R}$. For example, if a body-fixed coordinate system rotates around X_3 axis by angle θ, the direction cosine matrix will be:

$$\mathbf{R} = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2.4}$$

where l_{ij} ($i\ j$ =1,2,3) is the cosine of the angle between axis *i* of the new coordinate (body-fixed) and *j* axis of the old coordinate (moving coordinate) or the projection of *i* axis in the new coordinate on *j* axis of the old coordinate.

This matrix is the coordinate transforming matrix or rotation matrix $\mathbf{R}$ for the rigid body. It converts points in body-fixed coordinate to points in a global coordinate as follows:

$$x_i = \mathbf{R}X_i + x_i^o \tag{3.2.5}$$

Details on the rotation matrix can be found in Jia (1987) and Nikravesh (1988). The matrix $\mathbf{R}$ should be orthonormal, which means that each row should have unit length, and all rows are perpendicular to each other. To describe the angular orientation efficiently, the most common technique is to use Euler angles instead of using the nine elements of the matrix directly (Luo, et al. 2012). In Euler angle description, the rotation is decomposed into three elementary rotations of the body-fixed coordinate system relative to the moving coordinates. They represent three decomposed elementary rotations of a body-fixed coordinate system relative to the global system. However, using this description one degree of freedom may be lost in the case when one elementary rotation makes two axes to coincide. In such situations the effect of gimbal lock can occur. The process of gimbal lock is illustrated in Figure 3.3 to 3.6.

Assuming that a subject is represented by its body-fixed coordinate $OX_1X_2X_3$ and it is located in the origin of the global coordinate $oy_1y_2y_3$. A rotation described using

Euler angle includes three decomposed rotations around the three body-fixed coordinate axes respectively.

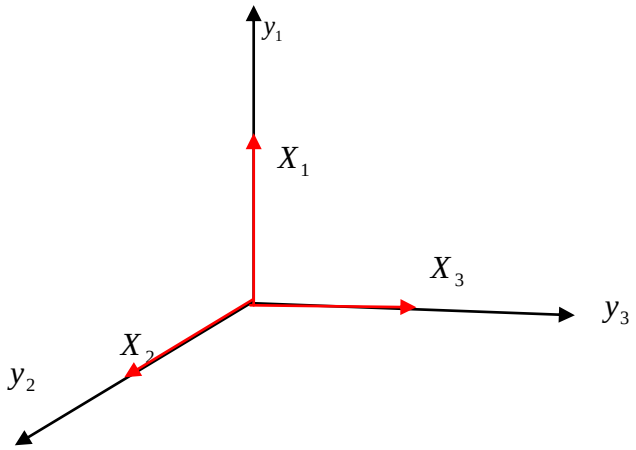


Figure 3.3: Initial orientation of an object represented by its body-fixed coordinate

First the object is rotated by an arbitrary angel, say -30^o around X_1 axis:

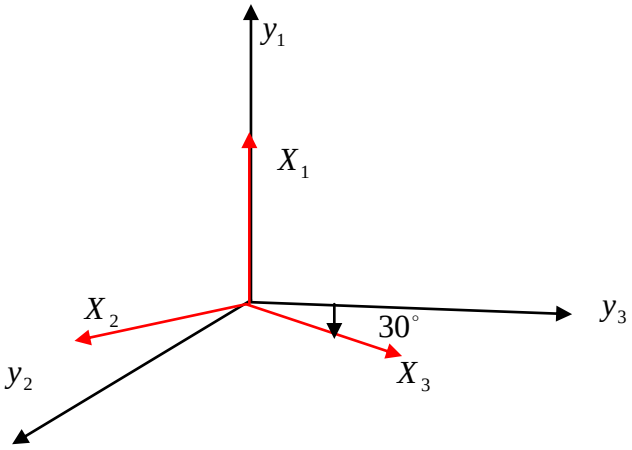


Figure 3.4: The object is rotated 30^o around X_1 axis

And then it is rotated by -90^o around X_2 axis

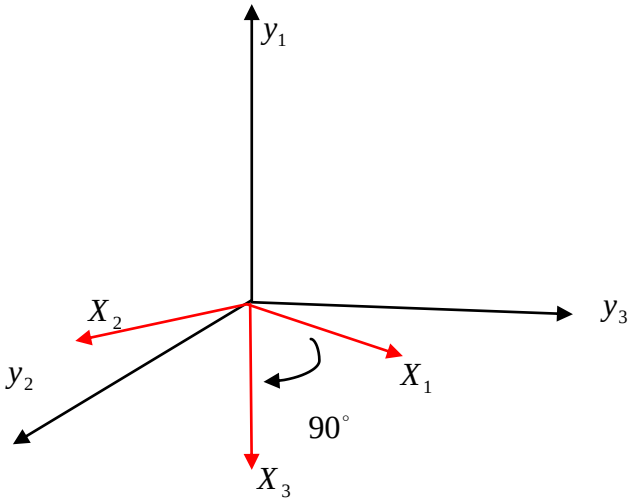


Figure 3.5: The object is rotated 90^o around X_2 axis

Now it can be seen that the current X_3 axis coincides the initial X_1 axis, in an opposite direction. Finally the object is rotated 40^o around this X_3 axis.

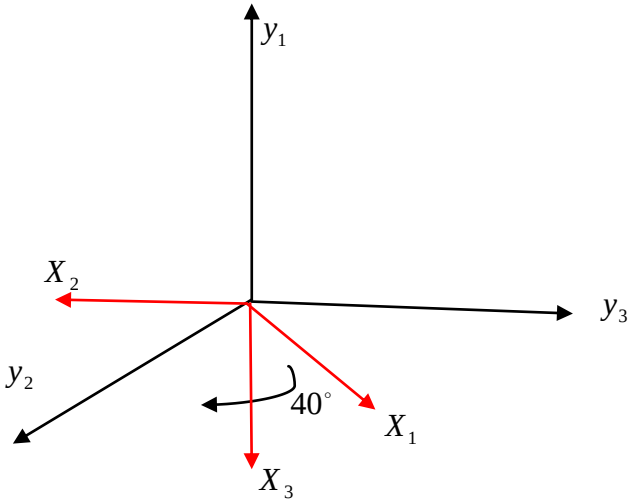


Figure 3.6: the object is rotated 40^o around X_3 axis

As the third rotation and the first rotation are about the same axis it is considered that one degree of freedom is lost.

To avoid this singularity and to ensure such motion is uniquely defined, Euler parameters are employed.

Euler parameters

According to Euler's Theorem, the general displacement of a body with a fixed point, i.e. angular movement, can be accomplished by a single rotation about a certain axis

with a finite angle. Therefore, it is reasonable to represent the coordinate transformation in terms of the parameters of this single rotation, i.e. the angle of rotation θ and a unit vector **e** of the rotation axis as shown in Figure 3.7. As the rotation can be defined by the coordinate transformation matrix or the parameters (θ, **e**), these two must be related to each other and the matrix can be derived based on (θ, **e**).

The Euler parameters

Euler vector: $$\mathbf{Q} = \mathbf{e}\sin\left(\frac{\theta}{2}\right) \tag{3.2.6}$$

Euler parameter: $$Q_0 = \cos\left(\frac{\theta}{2}\right)$$

$$\begin{cases} Q_1 = e_1 \sin\left(\frac{\theta}{2}\right) \\ Q_2 = e_2 \sin\left(\frac{\theta}{2}\right) \\ Q_3 = e_3 \sin\left(\frac{\theta}{2}\right) \end{cases} \tag{3.2.7}$$

Here, e_1, e_2 and e_3 are the projection of rotation axis in *x, y, z* axes, respectively.

The four parameters Q_0, Q_1, Q_2 and Q_3 satisfy the following equation:

$$Q_0^2 + Q_1^2 + Q_2^2 + Q_3^2 = 1 \tag{3.2.8}$$

which means that only three of them are independent. This indicates that the there are three rotational degrees of freedom. The derivation of the rotation matrix is shown below:

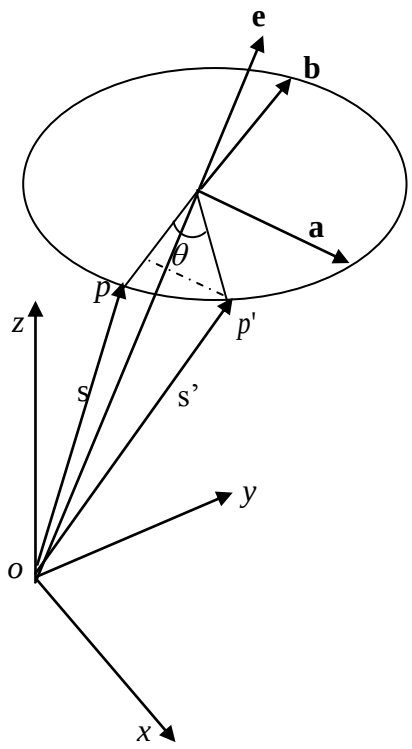


Figure 3.7: Rotation from p to p'

In Figure 3.7, the transformation of vector $\mathbf{s}$ into $\mathbf{s}'$ represents a rotation from p to p'. Assuming that vector $\mathbf{s}$ is fixed on the rigid body, and then $\mathbf{s}'$ can be expressed as

$$\mathbf{s}' = \mathbf{s} + (1 - \cos\theta)\mathbf{b} + \sin\theta\mathbf{a} \tag{3.2.9}$$

where $\mathbf{a}$ and $\mathbf{b}$ are perpendicular to each other and they have the same magnitude which equals to the distance from p to the rotation axis:

$$\mathbf{a} = \mathbf{e} \times \mathbf{s}\,,\ \mathbf{b} = \mathbf{e} \times \mathbf{a} \tag{3.2.10}$$

Substituting these into Equation (3.2.9), it can be re-written as:

$$\mathbf{s}' = \mathbf{s} + (1 - \cos\theta)\mathbf{e} \times (\mathbf{e} \times \mathbf{s}) + \sin\theta\mathbf{e} \times \mathbf{s} \tag{3.2.11}$$

Using Euler parameters given in Equation (3.2.6) and (3.2.7), the following equation can be obtained:

$$\mathbf{s'} = \mathbf{s} + 2\mathbf{Q} \times (\mathbf{Q} \times \mathbf{s}) + 2Q_0 \mathbf{Q} \times \mathbf{s} \tag{3.2.12}$$

And this can be converted this to a matrix expression as

$$\mathbf{s'} = \left(\mathbf{E} + 2\tilde{\mathbf{Q}}\tilde{\mathbf{Q}} + 2Q_0\tilde{\mathbf{Q}}\right)\mathbf{s} \tag{3.2.13}$$

where $\tilde{\mathbf{Q}}$ is the anti-symmetric matrix from $\mathbf{Q}$. As mentioned before, the rotation from s to s' can also be expressed with rotation matrix form

$$\mathbf{s'} = \mathbf{R} \cdot \mathbf{s} \tag{3.2.14}$$

Here, the rotation matrix $\mathbf{R}$ is obtained as:

$$\mathbf{R} = \begin{bmatrix} 1-2\left(Q_0^2+Q_1^2\right) & 2\left(Q_1Q_2 - Q_0Q_3\right) & 2\left(Q_1Q_3 + Q_0Q_2\right) \\ 2\left(Q_1Q_2 + Q_0Q_3\right) & 1-2\left(Q_0^2+Q_2^2\right) & 2\left(Q_2Q_3 - Q_0Q_1\right) \\ 2\left(Q_1Q_3 - Q_0Q_2\right) & 2\left(Q_2Q_3 + Q_0Q_1\right) & 1-2\left(Q_0^2+Q_3^2\right) \end{bmatrix} \tag{3.2.15}$$

The time derivative of the rotation matrix is related to the angular velocity as follows

$$\dot{\mathbf{R}} = \mathbf{R}\dot{\theta}_i \tag{3.2.16}$$

This can be expressed using the matrix form:

$$\frac{d}{dt}\begin{bmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -Q_1 & -Q_2 & -Q_3 \\ Q_0 & -Q_3 & Q_2 \\ Q_3 & Q_0 & -Q_1 \\ -Q_2 & Q_1 & Q_0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{3.2.17}$$

According to Equation (3.2.17), the parameters can be updated with the known angular velocity. To obtain angular velocity, equations of motion need to be solved. The governing equations for a rigid body motion will be discussed in the following section.

3.3 Governing equations of motion for rigid body

Since the motion of a particle of a rigid body is a combination of the translational motion of its mass centre (O) and a rotational motion about the mass centre, governing equations for these two motions are needed:

3.3.1 Translational motion of mass centre

For the translational motion, the Newton's second law can be applied:

$$m\ddot{u}_i^o = F_i \tag{3.3.1}$$

where *m* is the mass of the body, $\ddot{u}_i^o$ is the acceleration of the mass centre, F_i is the force applied on the centre of the mass (O). This equation can be re-written according to Figure 3.1 as

$$m\ddot{u}_i^o = \int_{\Omega_s} \rho_s \hat{F}_i d\Omega + \int_{S_T} \hat{T}_i dS + \int_{\sum} T_i dS \tag{3.3.2}$$

where $\hat{F}_i$ represents the body force, $\hat{T}_i$ and T_i are the traction forces on surface S_T and the interface $\sum$ respectively.

3.3.2 Rotation of the body about the mass centre

For rotational motion, the theorem of moment of momentum can be used:

$$\mathbf{I}\cdot\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}\times\mathbf{I}\cdot\dot{\boldsymbol{\theta}} = \mathbf{M} \tag{3.3.3}$$

where $\mathbf{I}$ is a second order tensor representing the moment of inertia, $\dot{\boldsymbol{\theta}}$ is a vector represented the angular velocity of the rigid body, $\ddot{\boldsymbol{\theta}}$ is the derivative of angular velocity with respect to time, $\mathbf{M}$ is a moment vector. Equation (3.3.3) given in the body-fixed coordinate system. In the principle inertia axis coordinate system of a rigid body, the equation of motion for rotation can be rewritten as (Fossen 2002; Jia 1987):

$$I_{xx}\ddot{\theta}_x + \left(I_{zz} - I_{yy}\right)\dot{\theta}_y\dot{\theta}_z = M_x$$

$$I_{yy}\ddot{\theta}_y + (I_{xx} - I_{zz})\dot{\theta}_z\dot{\theta}_x = M_y \qquad (3.3.4)$$
$$I_{zz}\ddot{\theta}_z + (I_{yy} - I_{xx})\dot{\theta}_y\dot{\theta}_z = M_z$$

The moment of inertia matrix is defined as follows:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \qquad (3.3.5)$$

The elements of this inertia matrix are calculated by summing up the contributions from all the particles of mass in the body. If the mass of each particle is δm, then these can be evaluated as:

$$I_{xx} = \sum \delta m(y^2 + z^2),\ I_{yy} = \sum \delta m(x^2 + z^2),\ I_{zz} = \sum \delta m(y^2 + x^2)$$
$$I_{xy} = \sum \delta m xy, \qquad I_{xz} = \sum \delta m xz, \qquad I_{yz} = \sum \delta m yz \qquad (3.3.6)$$

When it is necessary, this inertia matrix can be diagonalised using a transformation matrix made of its row Eigen vectors. This transformation matrix can be used to transform the coordinates of particles of the body and in this new coordinate system the inertia matrix will be diagonal.

With the angular velocity obtained from Equations (3.3.4) the Euler parameters can be updated. To solve Equation (3.3.4) the moment applied on the body needs to be known in a body-fixed coordinate system. Hence, if the moment is given in a global coordinate system then it needs to be transformed into a body-fixed coordinate system using the transpose matrix of $\mathbf{R}$:

$$\mathbf{M}_b = \mathbf{R}^T \cdot \mathbf{M}_g \qquad (3.3.7)$$

where $\mathbf{M}_b$ is the moment in body-fixed coordinate and $\mathbf{M}_g$ is the moment in global coordinate system. Sometimes moment needs to be calculated based on a known force. In this situation the force needs to be transformed to a body-fixed coordinate system first.

To summarise, the moment acting on the particles of rigid body needs to be expressed in the body-fixed coordinate system first. And from Equation (3.3.4) the angular velocity can be calculated. Then the rotation matrix can be derived from the angular velocity using Equation (3.2.17). Finally the new positions of particles in the global coordinate system can be obtained using Equation (3.2.5).

3.3.3 Boundary condition

In order to derive a unique solution from a set of partial differential equations, some conditions are required and boundary condition is one of those. There are generally three types of boundary conditions from the point view of mathematics: i.e. 1) Dirichlet boundary condition which specifies the values of the solution on the boundary of the domain, 2) Neumann boundary condition which specifies the values of the derivative of the solution on the boundary of the domain and 3) Cauchy boundary condition which is a mixed condition from Dirichlet boundary condition and Neumann boundary condition.

Physically, the boundary conditions can be classified based on the nature of the variables.

As shown in Figure 3.1, on a velocity boundary S_v of the rigid body, its velocity is assigned a prescribed value

$$v_i = \hat{v}_i \tag{3.3.8}$$

If this value is zero, it implies a fixed boundary. The displacement boundary can be ignored for rigid body since no deformation is expected.

On the force boundary S_T of the solid, its traction equals to a prescribed force $\hat{T}_i$.

3.4 Governing equation for fluid

Fluid can be treated as a continuum enclosed in a volume bounded by an arbitrary closed surface. To describe the motion of the continuum, the governing equations can be developed based on the principles of conservation of mass, conservation of momentum, and the laws of thermodynamics.

Let $\mathbf{x}$ denote the location of a particle, $\mathbf{v}$ represents the velocity. In the Lagrangian description, the velocity and acceleration of the particle are, respectively:

$$v_i = \frac{dx_i}{dt} \tag{3.4.1}$$

and

$$\frac{dv_i}{dt} = \frac{d^2 x_i}{dt^2} \tag{3.4.2}$$

Where the subscripts i and j indicates tensor index of value 1,2, or 3. The mass m contained in a domain V at time t is

$$m = \int_v \rho dV \tag{3.4.3}$$

where $\rho(\mathbf{x},t)$ is the density of the continuum at location $\mathbf{x}$ at time t. Conservation of mass requires $Dm/Dt = 0$, using Reynolds transport theorem the continuity equation can be derived

$$\int_v \frac{d\rho}{dt} dV + \int_s \rho \mathbf{v} \cdot \mathbf{n} dS = 0 \tag{3.4.4}$$

where $\mathbf{n}$ is the unit normal vector pointing outward from surface S. Since the results must hold for an arbitrary domain V, using the divergence theorem the integrand must vanish, i.e.

$$\frac{d\rho}{dt} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{3.4.5}$$

According to Newton's second law, the net force on a body is equal to its mass multiplied by the acceleration. The force can be considered as a combination of surface traction and body force f_i. Expressing the surface force in terms of a stress

vector σ_{ji}, the total force acting on the material occupying volume *V* interior to a closed surface S is

$$F = \int_s \sigma_{ji} n_j dS + \int_v f_i dV \tag{3.4.6}$$

According to Gauss's theorem, F can be expressed as:

$$F = \int_v \left(\frac{\partial \sigma_{ij}}{\partial x_j} + f_i \right) dV \tag{3.4.7}$$

For a unit mass, we have

$$\rho \frac{Dv_i}{Dt} = \frac{\partial \sigma_{ij}}{\partial x_j} + f_i \tag{3.4.8}$$

The governing equations for fluid include the conservation of mass and momentum. In a Lagrangian framework these can be written as follows

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{v} = 0 \tag{3.4.9}$$

The body force of the fluid is gravity force, and the stress tensor can be considered as a combination of pressure and viscous force and so:

$$\frac{D\mathbf{v}}{Dt} = \mathbf{g} + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} - \frac{1}{\rho} \nabla P \tag{3.4.10}$$

where t is the time, $\mathbf{g}$ is the gravitational acceleration, P is the pressure, $\boldsymbol{\tau}$ is viscous stress tensor and $\frac{D}{Dt}$ is the material derivative. The momentum equations include three forcing terms, i.e. body force, forces due to divergence of stress tensor and the

pressure gradient. For incompressible fluids, the mass density takes a constant, so that Equation (3.3.9) reduces to

$$\nabla \cdot \underline{v} = 0 \tag{3.4.11}$$

Assuming a Newtonian fluid, the viscous stress tensor $\boldsymbol{\tau}$ in the momentum conservative equation is related to the velocity as:

$$\tau_{ij} = \tau_{ji} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{3.4.12}$$

where μ is the dynamic viscosity coefficient. Hence Equation (3.3.10) can be written as

$$\frac{D\mathbf{v}}{Dt} = \mathbf{g} + \frac{\mu}{\rho}\nabla^2\mathbf{v} - \frac{1}{\rho}\nabla P \tag{3.4.13}$$

Using the chain rule, the gradients on the right hand-side of Equation (3.4.12) can be approximated as:

$$\left(\frac{\partial v_i}{\partial x_j}\right)_a = \left(\frac{v_a^i - v_b^i}{r_{ab}}\right)\left(\frac{x_a^j - x_b^j}{r_{ab}}\right) \tag{3.4.14}$$

Substituting Equation (3.4.14) into Equation (3.4.12), the formula for shear stress can be derived.

Boundary conditions

Boundary conditions for fluid can physically be classified as include inlet, outlet, free surface and solid-wall boundaries. On the free surface, the fluid motion should satisfy a kinematic condition

$$\frac{DY(x, y, z, t)}{Dt} = 0 \tag{3.4.15}$$

which implies that the free surface is a material surface. Here, Y is the function describing the height of the free surface. However, when fluid motion is given in Lagrangian form, this condition is automatically satisfied.

It is assumed that the pressure on the free surface is an atmospheric pressure of value zero as the reference pressure, so that the dynamic condition on the free surface is given by

$$P = 0 \tag{3.4.16}$$

On the velocity boundaries (inlet), velocity is assigned to be a prescribed value

$$\mathbf{v} = \hat{\mathbf{v}} \tag{3.4.17}$$

If this prescribed value is zero, this equation denotes a fixed fluid boundary condition. On the outlet boundary which is assumed to be infinite far away from the flow the pressure is assumed to be constant.

At the fixed solid boundary, no-slip condition is applied when the velocity of the fluid at the wall boundary is set to zero or free slip condition is applied when the tangential velocity of fluid is not zero but the normal velocity is zero. In our cases no slip boundary condition is applied.

3.5 Fluid solid interaction interface

On the wetted interface $\sum$, the motion of a solid particle should be coupled with a fluid particle and the following conditions are to be satisfied. To ensure no discontinuity on the fluid solid interface, the velocity of the fluid and the velocity of the solid are the same at each point, i.e.

$$v_i^f = v_i^s \tag{3.5.1}$$

It is necessary to satisfy the dynamic equilibrium condition at the interaction interface,

$$\tau_{ij}^f n_j + T_i = 0 \tag{3.5.2}$$

where τ_{ij}^{f} is the stress of the solid at interface, n_j is the unit vector in outer normal direction of the fluid boundary.

When the fluid stress on the solid equals zero, it indicates that the fluid is separated from the solid and in this situation Equation (3.5.1) is not required.

3.6 Summary

The governing equations used to model the fluid rigid body interaction are illustrated in Figure 3.8. Fluid motion is governed by N-S equations expressed in continuity equation and momentum conservative equation. Specific boundary conditions will be assigned to the particles at the right position. For rigid body, the motion is a combination of translation of mass centre and rotation around the mass centre. Translation of the mass centre (O) is simply governed by Newton's second law. For angular movement of particles in the solid, a transformation matrix in terms of quaternion is adopted. An overall chart of the content of this chapter is given below in Figure 3.8.

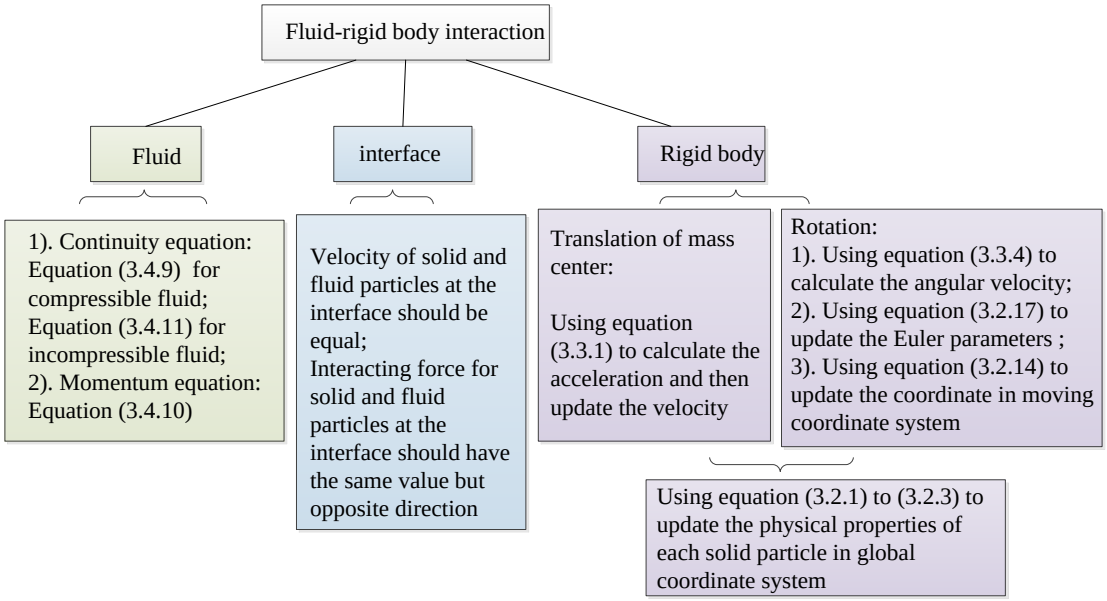


Figure 3.8: Governing equations for the motion of fluid rigid body interaction

Chapter 4 SPH formulation for nonlinear fluid-rigid body interactions

4.1 Basic SPH formulation

The basic formulation of SPH will be discussed in this section. As mentioned before, SPH is based on the theory of integral interpolant (Monaghan 1987; 1988; 1989; Monaghan & Kocharyan 1995; Liu & Liu 2003b), a general function $A(\mathbf{x})$ can be reproduced by an kernel approximation as $\langle A(\mathbf{x})\rangle$ (Monaghan 1982; Monaghan & Gingold 1983; Monaghan & Poinracic 1985a)

$$\langle A(\mathbf{x})\rangle = \int_{\Omega} A(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{4.1.1}$$

where $W(\mathbf{x}-\mathbf{x}')$ is the kernel function and h is the smoothing length which defines the influence domain of the kernel function. Similar process can be applied to the gradient of function approximation

$$\langle \nabla\cdot A(\mathbf{x})\rangle = \int_{\Omega} (\nabla\cdot A(\mathbf{x}'))W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{4.1.2}$$

In order to facilitate numerical approximation, the infinitesimal volume $d\mathbf{x}'$ in the integral equation (4.1.1) is replaced by the particle volume which can be expressed using mass m and density ρ,

$$dx' = \frac{m_b}{\rho_b} \tag{4.1.3}$$

The SPH particle approximation form can be derived if the integration is approximated by a summation over the neighbouring particles which are located within the smoothing length domain

$$\langle A(\mathbf{x})\rangle_{\mathrm{a}} = \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \frac{A_{\mathrm{b}}}{\rho_{\mathrm{b}}} W(|\mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\mathrm{b}}|, h) \tag{4.1.4}$$

The subscript a indicates the specific particle and b indicates neighbouring particles and N is the total number of particles inside the smoothing domain. Similarly, the approximation for spatial derivatives can be expressed as

$$\langle \nabla A(\mathbf{x})\rangle_{\mathrm{a}} = \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \frac{A_{\mathrm{b}}}{\rho_{\mathrm{b}}} \nabla W(\mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\mathrm{b}}, h) \tag{4.1.5}$$

The detailed derivation process is shown in the appendices. This equation implies that the derivatives of any function can be found by differentiating the kernel rather than by using grids. As a consequence, instead of solving partial differential equations for hydrodynamics problems, only ordinary differential equations need to be solved

The derivative of the kernel function can be expressed as

$$\nabla W_{\mathrm{ab}} = \frac{\mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\mathrm{b}}}{r_{\mathrm{ab}}} \frac{dW_{\mathrm{ab}}}{dr_{\mathrm{ab}}} \tag{4.1.6}$$

where r_{ab} is the distance between particle a and b, W_{ab} is the associated kernel function. From Equation (4.1.6) it is clear that the gradient of a function at particle a is approximated with a summation of the function values at each neighbouring particle times the gradient of the kernel function.

4.2 General SPH formulism for N-S equations

In WCSPH method, Equations (3.4.9) and (3.4.10) are used as governing equations. SPH formulation of these two equations can be derived simply by applying equation (4.1.4) to the right hand-side of Equation (3.4.9)

$$\frac{D\rho_a}{Dt} = -\rho_a \sum_{b=1}^{N} \frac{m_b}{\rho_b} v_b^j \frac{\partial W_{ab}}{\partial x_a^j} \tag{4.2.1}$$

The subscripts a, b represent different particles; superscripts *i*, *j* indicate different coordinate directions and *N* is the number of particles inside the smoothing domain, i.e. number of neighbouring particles. Another particle form of density gradient with respect to time can be derived by considering the following expressions

$$\begin{aligned} \int_\Omega \nabla W(\underline{x} - \underline{x}', h) dx' &= \nabla 1 \\ &= \sum_{b=1}^{N} \frac{m_b}{\rho_b} \frac{\partial W_{ab}}{\partial x_a^j} \\ &= 0 \end{aligned} \tag{4.2.2}$$

and adding Equation (4.2.2) to the right hand side of Equation (4.2.1) we can have the anti-symmetric formulation

$$\begin{aligned} \frac{D\rho_a}{Dt} &= \rho_a v_a^j \sum_{b=1}^{N} \frac{m_b}{\rho_b} \frac{\partial W_{ab}}{\partial x_a^j} - \rho_a \sum_{b=1}^{N} \frac{m_b}{\rho_b} v_b^j \frac{\partial W_{ab}}{\partial x_a^j} \\ &= \rho_a \sum_{b=1}^{N} \frac{m_b}{\rho_b} v_{ab}^j \frac{\partial W_{ab}}{\partial x_a^j} \end{aligned} \tag{4.2.3}$$

where

$$v_{ab}^j = v_a^j - v_b^j \tag{4.2.4}$$

Equation (4.2.3) uses the relative velocities of particle pairs in the smoothing domain and it is usually preferred.

If the density is continuous everywhere, the continuity equation can be represented by SPH in another form by considering:

$$-\rho \frac{\partial v^j}{\partial x^j} = -\left(\frac{\partial (\rho v^j)}{\partial x^j} - v^j \cdot \frac{\partial \rho}{\partial x^j} \right) \tag{4.2.5}$$

and converting the right hand side part into SPH form

$$\frac{\partial\left(\rho v^j\right)}{\partial x^j} - v^j \cdot \frac{\partial \rho}{\partial x^j} = \sum_{\mathrm{b}=1}^{N} \rho_{\mathrm{b}} v_{\mathrm{b}}^j \cdot \frac{\partial W_{ij}}{\partial x^j} - \sum_{\mathrm{b}=1}^{N} \rho_{\mathrm{b}} v_{\mathrm{a}}^j \cdot \frac{\partial W_{ij}}{\partial x^j} \tag{4.2.6}$$

By substituting Equation (4.2.6) into (4.2.5) and considering continuity Equation (3.4.5), the gradient of density with respect to time can be re-written as

$$\frac{D\rho_{\mathrm{a}}}{Dt} = \sum_{\mathrm{b}=1}^{N} \rho_b v_{ab}^j \cdot \frac{\partial W_{\mathrm{ab}}}{\partial x_{\mathrm{a}}^j} \tag{4.2.7}$$

Compared to Equation (4.2.3), Equation (4.2.7) has a simpler form and so it is more widely used. However, since the derivation of (4.2.7) is based on an assumption of continuous density, it is not appropriate for multi-phase flows especially when the density ratio is large and the interface is not specified. Instead, Equation (4.2.3) should be applied in this situation (Monaghan 2012; Sun, et al. 2012). An investigation of air water two-phase flow will be presented in Chapter 9. Different equations should be selected for different applications. In ISPH method if constant density is used all the equations to update density can be ignored.

The force term to update velocity in Equation (3.4.8) can be written in a symmetric form as

$$\begin{aligned}\left(\frac{1}{\rho}\nabla\sigma\right)_{\mathrm{a}} &= \frac{\partial}{\partial x_{\mathrm{a}}^j}\left(\frac{\sigma_{ij}}{\rho}\right) + \frac{\sigma_{ij}}{\rho^2}\frac{\partial \rho}{\partial x_{\mathrm{a}}^j} \\ &= \sum_{\mathrm{b}=1}^{N} \frac{m_{\mathrm{b}}}{\rho_{\mathrm{b}}} \frac{\sigma_{\mathrm{b}}^{ij}}{\rho_{\mathrm{b}}} \frac{\partial W_{\mathrm{ab}}}{\partial x_{\mathrm{a}}^j} + \frac{\sigma_{\mathrm{a}}^{ij}}{\rho_{\mathrm{a}}^2} \sum_{\mathrm{b}=1}^{N} \frac{m_{\mathrm{b}}}{\rho_{\mathrm{b}}} \rho_{\mathrm{b}} \frac{\partial W_{\mathrm{ab}}}{\partial x_{\mathrm{a}}^j} \\ &= \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \left(\frac{\sigma_{\mathrm{a}}^{ij}}{\rho_{\mathrm{a}}^2} + \frac{\sigma_{\mathrm{b}}^{ij}}{\rho_{\mathrm{b}}^2}\right) \nabla_j W_{\mathrm{ab}}\end{aligned} \tag{4.2.8}$$

This formula is the most widely used as it conserved angular and linear momentum because of the symmetry of the formula. However, similar to the density Equation (4.2.7), this formula also assumes continuous density so it is not suitable for multi-phase flow (Colagrossi & Landrini 2003; Sun, et al. 2012). Another form should be used in this situation which is:

$$\left(\frac{1}{\rho}\nabla\sigma\right)_{\mathrm{a}} = \frac{1}{\rho_{\mathrm{a}}}\sum_{\mathrm{b}=1}^{N}\frac{m_{\mathrm{b}}}{\rho_{\mathrm{b}}}\left(\sigma_{\mathrm{a}}^{ij}+\sigma_{b}^{ij}\right)\frac{\partial W_{\mathrm{ab}}}{\partial x_{\mathrm{a}}^{j}} \tag{4.2.9}$$

The problem from density discontinuity is eliminated in this form.

4.2.2 Viscosity effect

For many fluids, the stress tensor τ in the momentum conservative equation is related to the rate of strain as follows

$$\tau_{ij} = \tau_{ji} = \mu\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right) \tag{4.2.10}$$

where μ is the dynamic viscosity coefficient of the fluid.

Using chain rule for the RHS terms

$$\begin{aligned}\left(\frac{\partial v_i}{\partial x_j}\right)_{\mathrm{a}} &= \left(\frac{\partial v_i}{\partial r_{\mathrm{ab}}}\right)\left(\frac{\partial r_{\mathrm{ab}}}{\partial x_j}\right)\\ &= \left(\frac{v_{\mathrm{a}}^{i}-v_{\mathrm{b}}^{i}}{r_{\mathrm{ab}}}\right)\left(\frac{x_{\mathrm{a}}^{j}-x_{\mathrm{b}}^{j}}{r_{\mathrm{ab}}}\right)\end{aligned} \tag{4.2.11}$$

For fluid such as water, the viscosity coefficient has a constant value. So the SPH formulation of viscosity term can be written as (Shao & Lo 2003)

$$\left(\frac{1}{\rho}\nabla\cdot\boldsymbol{\tau}\right)_{\mathrm{a}} = \left(\frac{\mu}{\rho}\nabla^{2}\mathbf{v}\right)_{\mathrm{a}} = \sum_{\mathrm{b}=1}^{N}\frac{2m_{\mathrm{b}}\mu}{\left(\rho_{\mathrm{a}}+\rho_{\mathrm{b}}\right)^{2}}\frac{\mathbf{r}_{ab}\cdot\nabla W_{\mathrm{ab}}\left(\mathbf{v}_{a}-\mathbf{v}_{b}\right)}{\left|\mathbf{r}_{\mathrm{ab}}\right|^{2}+\varphi^{2}} \tag{4.2.12}$$

where φ is a small value used to prevent numerical divergence when two particles are too close to each other. Hence, normally Equation (4.2.8) is only applied to pressure.

Artificial viscosity

Sometimes an artificial viscosity is used to improve the numerical stability of SPH computation. This term is added in the momentum equations when calculating the

velocity to damp out some artificial oscillation for particles approaching each other (Johnson & Beissel 1996a).

The SPH formula for this term is as follows:

$$\Pi_{ab} = \begin{cases} \dfrac{-\alpha_{\Pi}\bar{c}_{ab}\phi_{ab} + \beta_{\Pi}\phi_{ab}^2}{\bar{\rho}_{ab}} & \mathbf{v}_{ab}\cdot\mathbf{x}_{ab} < 0 \\ 0 & \mathbf{v}_{ab}\cdot\mathbf{x}_{ab} > 0 \end{cases} \tag{4.2.13}$$

where

$$\phi_{ab} = \frac{h_{ab}\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}}{|\mathbf{x}_{ab}|^2 + \varphi^2} \tag{4.2.14}$$

$$\bar{c}_{ab} = \frac{1}{2}(c_a + c_b)$$

$$\bar{\rho}_{ab} = \frac{1}{2}(\rho_a + \rho_b) \tag{4.2.15}$$

$$h_{ab} = \frac{1}{2}(h_a + h_b)$$

$$\mathbf{v}_{ab} = \mathbf{v}_a - \mathbf{v}_b, \mathbf{x}_{ab} = \mathbf{x}_a - \mathbf{x}_b$$

where $\alpha_{\Pi}, \beta_{\Pi}$ are constants that are typically set around 1.0 (Rabczuk, et al. 2006). The factor φ (namely, $\varphi = 0.1h_{ab}$) is added to prevent numerical divergence when two particles are getting too close. c and v represent the speed of sound and the particle velocity vector respectively.

4.2.3 Pressure calculation

From the momentum equation, it can be seen that the forces acting on fluid particles are from pressure, viscosity and gravity. Gravity is known whereas viscosity sometimes can be ignored. Hence, in order to determine the motion in a fluid, the most important factor is the pressure. To calculate pressure of fluid with SPH one can use weakly compressible SPH (WSHP) or truly incompressible SPH (ISPH) as discussed in Chapter 2 and the algorithm process of these two methods will be illustrated here.

Pressure calculation in WCSPH

Assuming that the fluid is slightly compressible (Monaghan 1994) and Mach number is sufficiently small so that the density fluctuations is less than 0.01, the pressure can be calculated using the equation of state (Batchelor 1973).

$$P = B\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) \tag{4.2.16}$$

where γ is the polytrophic constant, normally chosen as 7 for incompressible fluid (water) and 1.4 for compressible fluid (air). ρ_0 is the initial fluid density and B is a constant that can be calculated as:

$$B = \frac{c_0^2 \rho_0}{\gamma} \tag{4.2.17}$$

where ρ_0 and c_0 denote the reference density and a numerical speed of sound in the fluid, respectively. The influence of pressure on the velocity can be computed by substituting the new pressure into SPH formula (4.2.8) or (4.2.9).

This weakly compressible SPH is easy to complement as explicit algorithm is used (Monaghan 1994; Morris, et al. 1997; Hu & Adams 2006). However, since the pressure value depends strongly on the fluctuation of density, it lacks accuracy for pressure calculation.

Pressure calculation in ISPH

Another way of computing pressure is to treat the fluid as truly incompressible and a then a Poisson's equation needs to be solved to obtain the pressure values (Pozorski & Wawrenczuk 2002). This can be done by enforcing the velocity divergence free and zero density variation conditions (Shao & Lo 2003; Hosseini, et al. 2007), or by using constant density (Lee, et al. 2010). The incompressible Navier-Stokes Equations (3.3.10) and (3.3.11) are used to describe the motion of water. The momentum equation is split into two parts to derive the equation for pressure values. The first part considers the effect from body force and viscosity

$$\mathbf{v}^{*} = \mathbf{v}^{n} + \mathbf{g}\delta t + \left(\frac{1}{\rho}\nabla \cdot \boldsymbol{\tau}^{n}\right)\delta t \tag{4.2.18}$$

The second part considers the effect from pressure

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^{*}}{\delta t} = -\frac{1}{\rho}\nabla P^{n+1} \tag{4.2.19}$$

Taking divergence of Equation (4.2.19) and substituting the outcome into incompressible continuity Equation (3.4.11) the Poisson's equation for pressure can be derived

$$\Delta P^{n+1} = \frac{\rho}{\delta t}\nabla \cdot \mathbf{v}^{*} \tag{4.2.20}$$

And this Poisson's equation is then converted to SPH formulation:

$$\sum_{\mathrm{b}=1}^{N} \frac{m_{\mathrm{b}} P_{\mathrm{ab}}^{n+1} \mathbf{r}_{\mathrm{ab}}^{n}}{r_{\mathrm{ab}}^{2} + \eta^{2}} \cdot \nabla W_{\mathrm{ab}}^{n} = -\frac{\rho}{2\delta t} \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \mathbf{v}_{\mathrm{ab}}^{*} \cdot \nabla W_{\mathrm{ab}}^{n} \tag{4.2.21}$$

Equation (4.2.21) can be solved implicitly using for example the Bi-CGSTAB method (Vorst 1992). Or it can be solved explicitly as (Hosseini, et al. 2007)

$$P_{\mathrm{a}}^{n+1} = \frac{-\frac{\rho}{2\delta t} \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \mathbf{v}_{\mathrm{ab}}^{*} \cdot \nabla W_{\mathrm{ab}}^{n} + \sum_{\mathrm{b}=1}^{N} \frac{m_{\mathrm{b}} P_{\mathrm{b}}^{n} \mathbf{r}_{\mathrm{ab}}^{n}}{r_{\mathrm{ab}}^{2} + \eta^{2}} \cdot \nabla W_{\mathrm{ab}}^{n}}{\sum_{\mathrm{b}=1}^{N} \frac{m_{\mathrm{b}} \mathbf{r}_{\mathrm{ab}}^{n}}{r_{\mathrm{ab}}^{2} + \eta^{2}} \cdot \nabla W_{\mathrm{ab}}^{n}} \tag{4.2.22}$$

Time step sizes allowed for implicit solution and explicit solution are similar. Using explicit solution approach can reduce computational time but the accuracy of the algorithm cannot be guaranteed for all cases as no iteration is performed. Comparison of the performance of these two solution approaches will be given later in Chapter 6.

Finally, the velocity of each fluid particle can be renewed for the next time step as

$$\begin{aligned}\mathbf{v}_{\mathrm{a}}^{n+1} &= \mathbf{v}_{\mathrm{a}}^{*} - \frac{1}{\rho}\nabla P^{n+1}\delta t \\ &= \mathbf{v}_{\mathrm{a}}^{*} - \delta t\sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}}\left(\frac{P_{\mathrm{a}}^{n+1}}{\rho_{\mathrm{a}}^{2}} + \frac{P_{\mathrm{b}}^{n+1}}{\rho_{\mathrm{b}}^{2}} + \Pi_{\mathrm{ab}}\right)\cdot\nabla W\left(\mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\mathrm{b}}, h\right)\end{aligned} \tag{4.2.23}$$

The new positions of each particle is then updated based on the velocities

$$\mathbf{x}^{n+1} = \mathbf{x}^{n} + \mathbf{v}^{n+1}\delta t \tag{4.2.24}$$

Pressure calculation is one of the most important parts of the algorithm since the motion is driven largely by the pressure force. Using WCSPH method, the pressure can be calculated easily based on the density variations but this lacks accuracy since it is sensitive to the density change. Extra correction is necessary for this method. With ISPH method pressure can be obtained more accurately but more computational time is needed for the solution of Poisson's equation. The performance of these two methods will be also compared in Chapter 6.

The overall algorithm processes for both WCSPH and ISPH are the same as shown in Figure 4.1.

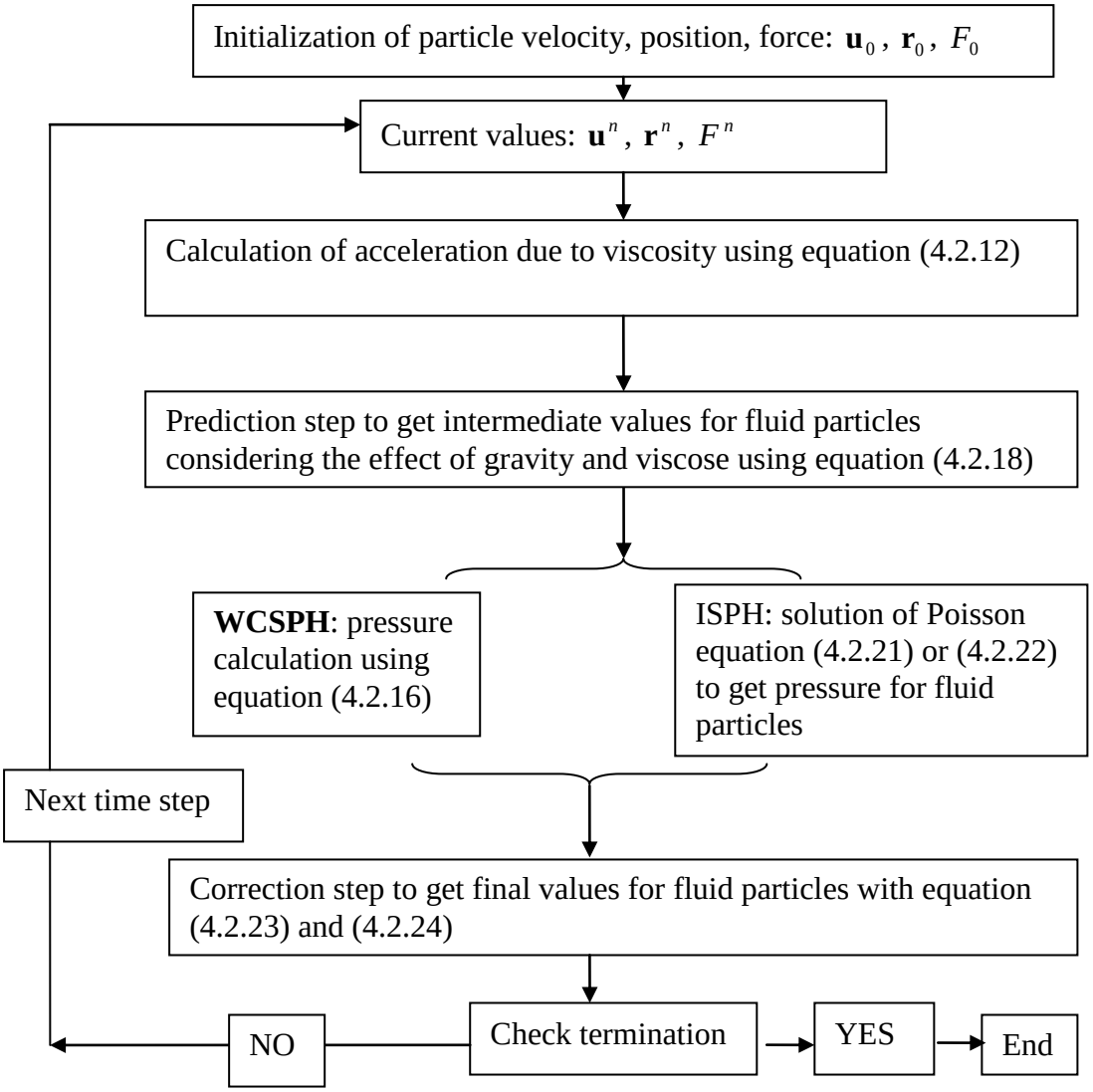


Figure 4.1: Flowchart of numerical algorithm for fluid

4.3 SPH formulation for solid

The solid particles on the interface are treated as boundaries for the fluid. The pressure values of the solid particles are obtained directly from the Poisson's equation and they are considered as external force acting on the solid. Because of the numerical property of SPH method, the fluid and solid motion will be coupled automatically in the algorithm without the conventional interface conditions mentioned in section 3.4.

In order to solve the governing equations of sold with the particles based system, all the forcing terms in Equation (3.3.2) should be expressed in particle formulae. At

time step $n+1$, if $F_i^{B,n+1}, F_i^{s,n+1}, F_i^{I,n+1}$ are used to represent the total body force, total surface force and total fluid structure interaction force, respectively

$$F_i^{B,n+1} = \int_{\Omega_s} \rho_s \hat{F}_i^{n+1} d\Omega = \sum_{\mathrm{a}}^{N} \rho_{\mathrm{a}} \hat{F}_{\mathrm{a},i}^{n+1} \mathbf{r}_{\mathrm{ad}}^n \cdot \left(\mathbf{r}_{\mathrm{ab}}^n \times \mathbf{r}_{\mathrm{ac}}^n \right) \tag{4.3.1}$$

$$F_i^{s,n+1} = \int_s \hat{T}_i^{n+1} dS = \sum_{\mathrm{a}}^{N} \hat{T}_{\mathrm{a},i}^{n+1} \left| \mathbf{r}_{\mathrm{ab}}^n \times \mathbf{r}_{\mathrm{ac}}^n \right| \tag{4.3.2}$$

$$F_i^{I,n+1} = \int_{\sum} T_i^{n+1} dS = \sum_{\mathrm{a}}^{N} T_{\mathrm{a},i}^{n+1} \left| \mathbf{r}_{\mathrm{a\,b}}^n \times \mathbf{r}_{\mathrm{ac}}^n \right| \tag{4.3.3}$$

where *N* is the total number of solid particles. a, b, c and d are indexes of adjacent particles as shown in Figure 4.2. r_{ab} is the vector from particle a to b and $\left| \mathbf{r}_{\mathrm{ab}}^n \times \mathbf{r}_{\mathrm{ac}}^n \right|$ is equivalent to the area of the quadrilateral defined by particles a, b and c.

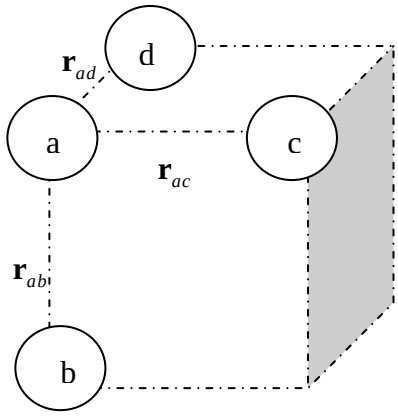


Figure 4.2: Adjacent particle in a unit volume of solid

In our case, $\hat{F}_i^{n+1}$ is from the gravity (g=9.81kg/m^2). And it is assumed that there is no surface traction so $F_i^{s,n+1}$ equals zero. The interaction force includes contribution from pressure and viscous forces. Viscous force F_i^{v} is computed based on the acceleration obtained due to viscosity effect.

$$F_i^{v,n+1} = \sum_{a}^{N} m_a \left(\frac{1}{\rho} \nabla \cdot \boldsymbol{\tau}^n \right) \tag{4.3.4}$$

And,

$$T_i^{n+1} = P^{n+1} n \cdot i + F_i^{v,n+1} \tag{4.3.5}$$

where n is the normal direction of the area on which the pressure is acting.

Similarly for the momentum Equation (3.4.5) or (3.4.6), if $M_i^{B,n+1}$, $M_i^{s,n+1}$, $M_i^{I,n+1}$ are used to represent the moments due to total body force, total surface force and total fluid structure interaction force at time $n+1$ respectively, then

$$M_i^{B,n+1} = \int_{\Omega_s} \rho_s e_{ijk} X_j^n \times F_k^{n+1} d\Omega = \sum_{\mathrm{a}}^{N} \rho_{\mathrm{a}} e_{ijk} X_{\mathrm{a},j}^n \times F_{\mathrm{a},k}^{n+1} \mathbf{r}_{ad}^n \cdot \left(\mathbf{r}_{ab}^n \times \mathbf{r}_{ac}^n \right) \tag{4.3.6}$$

$$M_i^{s,n+1} = \int_S e_{ijk} X_j^k \times \hat{T}_k^{n+1} dS = \sum_{\mathrm{a}}^{N} e_{\mathrm{ijk}} X_{\mathrm{a},j}^n \times \hat{T}_{\mathrm{a},k}^{n+1} \left| \mathbf{r}_{ab}^n \times \mathbf{r}_{ac}^n \right| \tag{4.3.7}$$

$$M_i^{I,n+1} = \int_{\sum} e_{ijk} X_j^k \times T_k^{n+1} dS = \sum_{\mathrm{a}}^{N} e_{\mathrm{ijk}} X_{\mathrm{a},j}^n \times T_{\mathrm{a},k}^{n+1} \left| \mathbf{r}_{ab}^n \times \mathbf{r}_{ac}^n \right| \tag{4.3.8}$$

Since the moment is calculated relative to the mass centre so $M_i^{B,n+1}$ is zero, and surface force is assumed to be zero as well, the only moment left is due to the pressure and viscous force from fluid $M_i^{I,n+1}$.

Hence, based on Equations (3.2.1) and (3.2.6), the translational and rotational acceleration of the mass centre can be updated.

4.4 Summary

N-S equations describing fluid motion are represented in SPH forms. In the simulations, density needs to be updated using Equation (4.2.7) for single phase flow or (4.2.3) for multi-phase flow if WCSPH method is used. Pressure can be calculated by the equation of state (4.2.16) and then substituted into (4.2.8) for pressure effect on the velocity in the situation of single phase and (4.2.9) for multi-phase flow. Together with the viscous force effect computed by (4.2.12) the acceleration can be obtained and then the velocity and position can be updated for the next time step. In ISPH

method, density is constant and pressure is calculated using Poisson's equation (4.2.22). Similar to WCSPH, pressure values is substituted into Equation (4.2.8) for single phase flow and (4.2.9) for multi-phase flow. A scheme to be used for multi-phase flow is to use WCSPH for compressible phase (air) and ISPH for incompressible flow (water) and this will be discussed in detail in Chapter 9. The total force and moment can be calculated by Equation (4.3.1) to (4.3.8). The interaction force term has two parts: one is pressure force and another is viscosity force. Pressure on the interface particles are calculated through Poisson's equation directly as these solid particles are treated as boundary particles for fluid. Viscous force is computed based on the acceleration obtained due to the viscosity effect. All the kinematic properties of solid particles such as velocity and position are used in the physical property calculations of fluid particles as they are neighbouring particles so that the coupling at the interface is automatically accounted for. The overall SPH representation of fluid rigid body interaction is illustrated in Figure 4.3.

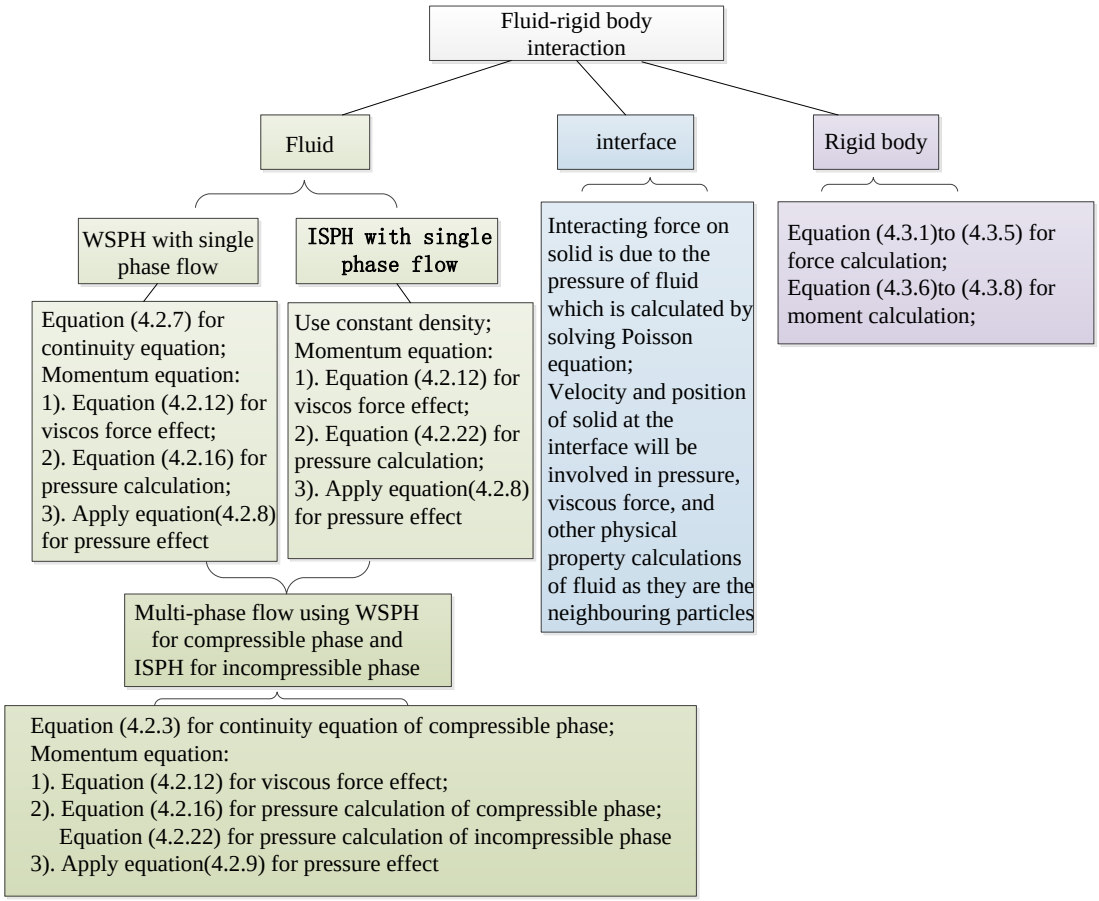


Figure 4.3: SPH representation of fluid rigid body interaction

Chapter 5 Implementation of SPH algorithm

In SPH method, the system is represented by a set of particles. These particles possess material properties and move according to physical laws. The property calculation of each particle is carried out by smoothing over neighbouring particles within the smoothing domain. The smoothing domain is the influence domain of the kernel function as explained in section 2.4. Because of the compact support property, the value of the kernel function outside the influence domain is zero. Hence, it is expected that no particle outside the smoothing domain should contribute to the approximation in particle form.

For the particles near boundaries, they do not have enough neighbouring particles as there is no particle outside the boundary. Special treatment on the boundary is necessary to prevent a particle from penetrating the boundaries and it is also important sometimes to ensure a correct calculation of flow parameters.

It is necessary to identify the neighbouring particles for each particle before any calculation can be conducted. Since the particles are arbitrarily distributed, neighbouring particle identification needs to be carried out for each particle at every time step. Searching for neighbouring particles is the most time consuming operation in the computation process. A proper searching algorithm is required to ensure an efficient simulation.

In this chapter, the important factors of SPH implementation including boundary treatments, computational strategies and time stepping algorithm will be discussed.

5.1 Boundary treatments

Boundary condition implementation is an important aspect of SPH implementation. The function may be incorrectly calculated using the particle approximation due to the absence of particles beyond the boundary, and instability may also occur in the evolution calculation for the function (Belytschko, et al. 1996).

Since the integral is approximated by a summation over the smoothing domain, a complete smoothing domain is important for a correct approximation. The particles within a smoothing length distance away from the boundary normally have a complete smoothing domain. However, for those particles near a boundary, the

smoothing domain is truncated and there are no particles outside the boundary as shown in Figure 5.1 (Liu, et al. 2003a), so the kernel condition cannot be satisfied.

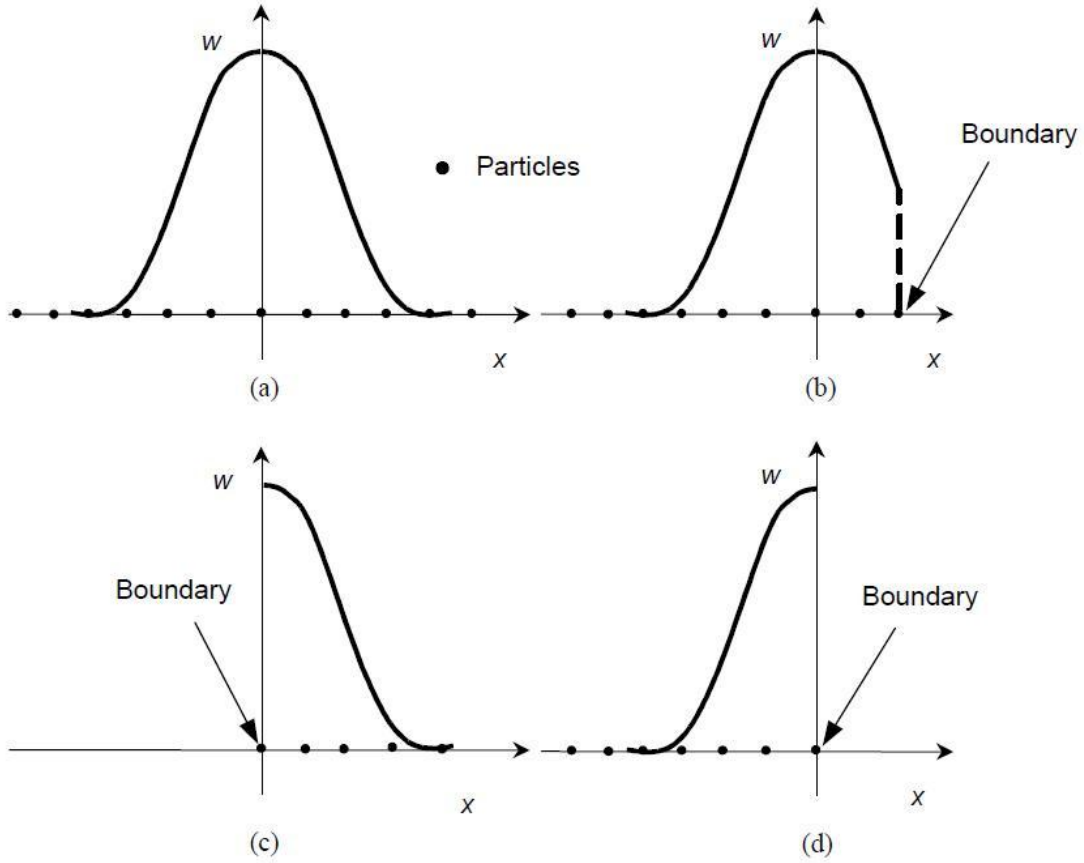


Figure 5.1: Kernel function for particles far away from boundaries (*a*), particles near the boundaries (*b*) and particles on the boundaries (*c*,*d*)

Special treatment for these particles near boundary is required in SPH method to resolve this problem.

Boundaries can belong to a solid or a fluid. Solid boundaries can be fixed or moving, fluid boundaries can be free surface, inlet or outlet. As the fluid is normally assumed to be confined the inlet and outlet boundary will not be considered.

Free-surface

The free surface conditions expressed in Chapter 3 should be satisfied and particles on the free surface need to be identified. In SPH method, the pressure of free surface particles is set to be zero to simplify the dynamic surface boundary conditions (Monaghan 1989). The following quantity is calculated to identify the free surface particles

$$\nabla \cdot r_a = \sum_{b=1}^{N} \frac{m_b}{\rho_b} \mathbf{r}_{ab} \cdot \nabla W_{ab} \tag{5.1.1}$$

This value should equal to 2 in 2D applications or 3 in 3D cases when the smoothing domain is not truncated but it is far below these values for free surface particles. So this value is 1.6 and 2.5 in 2D and 3D cases respectively.

Wall boundary

One of the drawbacks of SPH modelling is characterized by particle penetration of the wall. The solid walls are represented by particles which prevent the inner particles from penetrating the wall. Generally, there are three different schemes to achieve this: 1) mirror particles (Cummins & Rudman 1999) ; 2) repulsive forces (Monaghan 1994) or 3) dummy particles (Shao & Lo 2003; Crespo, et al. 2007).

Mirror particles and dummy particles are similar and they are called ghost particles in general. They are artificially particles placed outside the boundary. Mirror particles are generated at every time step. The boundary is treated as a mirror and when an inner fluid particle approaches the boundary a pseudo particle is generated on the other side of the boundary. This virtual particle has the same density and pressure but opposite velocity as the associated real particle (Randles & Libersky 1996). Mirror particles are more computationally time consuming they are not widely used. Alternatively, the artificial particles with fixed positions can be distributed in a staggered grid outside the system and they can be included in equations of continuity and state (Dalrymple & Knio 2001). These fixed dummy particles are commonly used.

Repulsive force is conventionally used in WCSPH method. The wall boundaries are modelled by a set of particles with fixed positions and zero velocities which have a repulsive force with the form of Lennard-Jones potential to the approaching inner fluid particles, as shown in Figure 5.3. This repulsive force can be calculated by

$$f(r) = D\left(\left(\frac{r_0}{r}\right)^{p_1} - \left(\frac{r_0}{r}\right)^{p_2}\right)\frac{\mathbf{r}}{r^2} \tag{5.1.2}$$

where D is a problem dependent constant and its value can be determined using other parameters of the problem. For example, one suggestion is to use $D = 5gH$, where H is the water depth (although there must be some difference between the bottom particles and the side particles D is taken as a unique value all through the particles to keep a simple algorithm) , the parameters $p_1 = 4$ and $p_2 = 2$ (Monaghan 1994). r is the distance from an inner particle to a boundary particle and r_0 is the cut-off distance normally selected to be the initial particle spacing. The value of $f(r)$ is set to be zero when $r > r_0$ so that the force is purely repulsive. This force can also be used as the repulsive force between different material particles for fluid flowing through a porous media (Monaghan & Kos 1999; Jiang, et al. 2007).

In the conventional ISPH method (Shao & Lo 2003), densities still need to be updated at every time step and these intermediate densities will be substitute into the Poisson's equation, a complete kernel domain is necessary to ensure a correct particle approximation hence ghost particles which mirror the physical properties of inner fluid particles are the usual treatment of wall boundary conditions (Lee, et al. 2008). Ghost particles are used to maintain an un-truncated kernel domain for the inner particles near the wall. Therefore, the consistence of SPH simulation near wall boundaries is ensured and the physical properties such as density can be calculated correctly.

However, when dealing with problems with complex solid boundaries the ghost particle boundary treatment becomes difficult to apply. Taking compartment flooding as an example, here water can fill both inside and outside the structure and at least two layers of ghost particles are needed, one on the inside wall and one on the outside wall respectively. These ghost particles sometimes overlap the true fluid particles. This can lead to inaccurate neighbouring particle counting and hence result in wrong predictions. It is also difficult to use ghost particles for curved boundary. Special consideration is required to calculate the exact position of the ghost particle for the points on the curved boundary since the position of the ghost particle is important to prevent particle penetration of the wall. If constant density is used for ISPH method which means density fluctuation is avoided, the boundary treatment can be simplified compared to the conventional ISPH method. These boundary treatments, i.e. using repulsive boundary force and denser wall particles on the boundary, have not been

used for ISPH method in the known literature although they have been used in WCSPH method previously. Practically, as long as the density can be kept as a constant, the main function of the boundary particles is to prevent inner particles from penetrating the walls. Therefore, repulsive force can be applied on the wall particles instead of using several lines of dummy particles which not only increases computational time but also complicates the model set-up especially in fluid structural interaction problems.

Another boundary treatment using denser wall particles is also a possibility. With repulsive force, all the particles can be maintained in a uniform arrangement but the additional force may disturb the pressure values on the boundary particles. This problem can be overcome by using denser wall particles with, say, half spacing of the inner fluid particles. These two boundary treatments can be chosen according to different situations. Both the boundary treatments allow efficient simulations with complex solid boundaries and they simplify the coupling approach for fluid structural interactions (Sun, et al. 2011).

Wall particles are used in the Poisson equation for pressure calculation. Using denser particles on the wall boundary can provide sufficient pressure to keep the inner particles away from the boundary. A halved spacing is set for the wall particles compared with the inner fluid particles as shown in Figure 5.2.

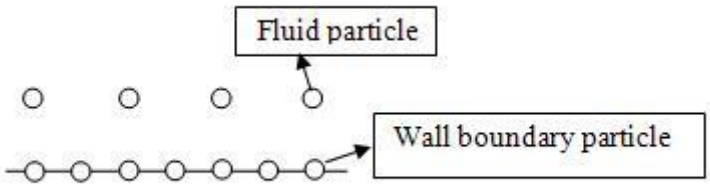


Figure 5.2: Boundary treatment: using half spacing on wall particles

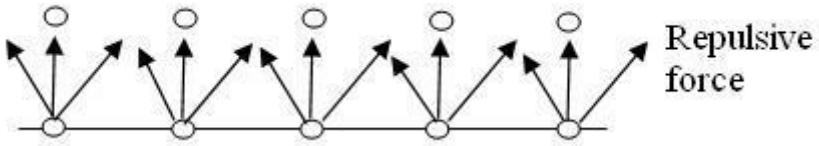


Figure 5.3: Boundary treatment: using repulsive force

5.2 Computational strategies

In SPH method, calculations of physical properties such as forces, velocity and densities are carried out for each particle based on summation over neighbouring particles located inside a cut-off radius r_c (r_c is linear proportional to the smoothing length). Identification of neighbouring particles must be accomplished before the solution of the governing equations.

Searching neighbouring particles for each particle by computing the distances between other particles in the system can be time consuming, especially when a large number of particles are used. Therefore, it is necessary to adopt an efficient algorithm to search for the neighbouring particles.

Two main methods are used to reduce the unnecessary computation of the distances: the first is to store dynamically the neighbourhood list of each particle (Verlet list) and the second one is to use a framework of fixed cells (cell-linked list) (Viccione, et al. 2008). There are a few other methods that are used to improve the searching efficiency such as oct-tree methods that are used mostly in astrophysical problems (Stellingwerf & Wingate 1994). An algorithm combining Verlet list and cell-linked list has been used (Yao, et al. 2004; Dominguez, et al. 2010). For particles with variable cut-off distances the search methods of cell list and oct-tree can be combined (Awile, et al. 2012).

In this project, this combined list is adopted and the computational time is further reduced by making use of the symmetrical characteristic of neighbouring particles.

5.2.1 Cell-Linked List algorithm

Since the neighbouring particles are located within the smoothing domain, it is beneficial to divide the space into a number of regions (cells) to improve the search efficiency since neighbouring particles will only exist in the neighbouring regions. In the application of cell-linked list algorithm, first the problem domain is first partitioned into many regular cells, and every particle is assigned to a cell according to its position. The size of the cell can be chosen to be the same as the cut-off distance or slightly larger. Since the neighbours of each cell is known and fixed, and for each particle, only its own cell and the neighbouring cells need to be searched thus

substantial savings of computational time can be achieved (Hockney & Eastwood 1981; Monaghan & Gingold 1983).

The implementation of this process can be outlined as follows:

1). Divide the problem domain into N_x, N_y, N_z cells in x, y, z direction. Here a 2D example is shown in Figure 5.4.

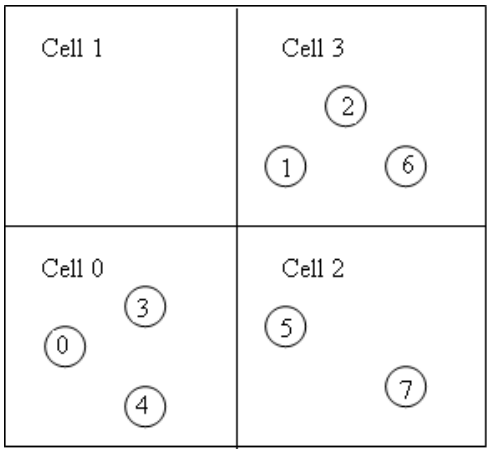


Figure 5.4: Discretizing the domain into cells and storing the particles in appropriate cell

The number of cells in x, y and z directions are:

$N_x = L_x / r_{cx}, N_y = L_y / r_{cy}, N_z = L_z / r_{cz},$

here L_x, L_y and L_z are the domain lengths, r_{cx}, r_{cy} and r_{cz} are the cell sizes,

N_x, N_y and N_z are the number of cells in x, y, z direction respectively.

The cell size can be adjusted to make the number of particles processed in the searching procedure as small as possible (Allen & Tildesley 1990; Mattson & Rice 1999). Constant cut-off radius is used through the algorithm since it is more efficient than using a constant number of neighbour particles (Wróblewski, et al. 2007).

For a certain cell if the index along each direction is N_{cx}, N_{cy}, N_{cz}.

Then it is identified as

$C = N_{cx} \times N_y \times N_z + N_{cy} \times N_z + N_{cz}$.

For a given particle a, its cell indices can be calculated as

$$N_{cx} = \text{int}\left(x[a]/r_{cx}\right), \quad N_{cy} = \text{int}\left(y[a]/r_{cy}\right), \quad N_{cz} = \text{int}\left(z[a]/r_{cz}\right)$$

2). Store each particle in an appropriate cell according to the positions. Two lists will be needed to identify particles inside one cell: 1) $lscl[N_p]$ (N_p is the total number of particles) is used for particle a pointing to particle b and 2) $head[N_c]$ (N_c is the total number of cells) is used for storing the last particle of the cell.

An example of the linked-list ($lscl$) of particles is shown in Figure 5.5

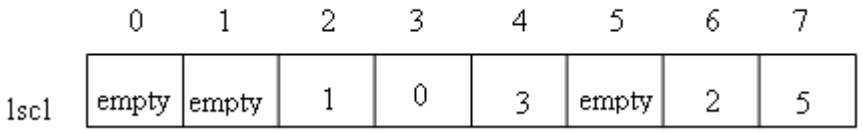


Figure 5.5: Example of the linked list array

This example is related to the particle distribution shown in Figure 5.4. Here, particle 0 points to empty as there is no particle before it in cell 0; particle 1 points to empty as there is no particle before it in cell 1; particle 2 points to particle 1 in cell 3 and so on.

The $head$ list stores the last particle of each cell is shown in Figure 5.6

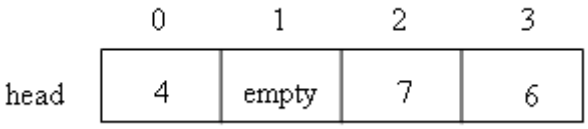


Figure 5.6: Example of head list for each cell

Taking cell 3 for example, since $head[3] = 6$; particles inside cell 3 can be reached using particle linked list starting from the head of the cell which is particle 6:

$lscl[6] = 2 \rightarrow lscl[2] = 1 \rightarrow lscl[1] = empty$

3). Calculate the distances between a particle and other particles in the cell itself and the adjoining cells so neighbouring particles can be identified and then the physical calculation can be carried out. This can be done either by storing the neighbouring cells in an array for each cell (with dimension of $9 \times N_c$) or computing the

neighbouring cell index during the solution process (cost $9 \times N_c$ times computation for every solution loop);

Summary

In this method, at least one array with size of N_c and one array with size of N_p need to be built for the neighbouring particle searching algorithm. The neighbouring list for each particle is not recorded. This neighbouring particle searching needs to be done at every time step. The situation will be worse if the physical properties need to be calculated in separate computation loops and this means that the same searching operation needs to be carried out several times within one time step. For example when an algorithm, such as predictive-corrective algorithm, uses two or more sub steps in one time step, the neighbouring particles need to be searched in every sub step before the change of particle positions. It is a waste of computational time to repeat the search process since the same neighbouring particles are used so the neighbouring list of each particle should be recorded.

5.2.2 Verlet List algorithm

The aim of Verlet list algorithm is to reduce the redundant distance computation by building a neighbour list which can be used for several time steps (N_m time steps). In order to do so, a radius r_m slightly larger than the cut-off distance is used to identify the potential neighbouring particles. It looks like a "skin" outside the cut-off region as shown below in Figure 5.7

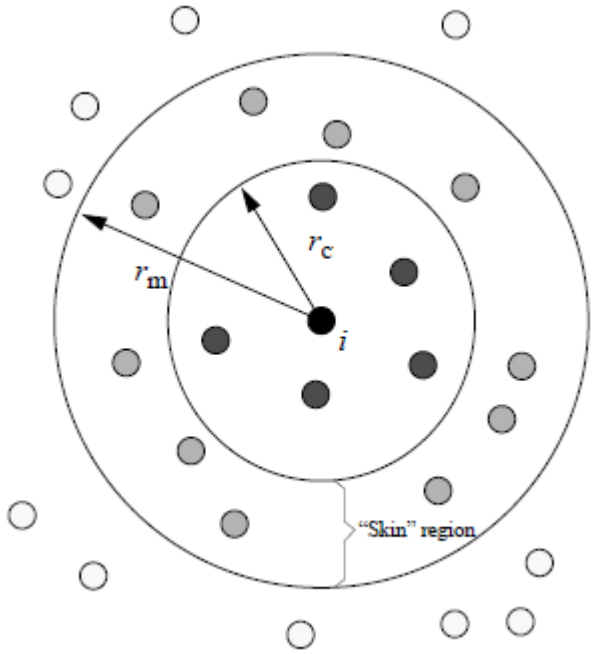


Figure 5.7: Influencing domain for Verlet list algorithm

r_m and N_m are chosen such that

$r_m - r_c > N_m v_{\max} \delta t$

where $v_{\max}$ is the maximum velocity and δt is the time step size.

The neighbouring particle list can be stored in a one dimensional array called *neighbourlist* with dimension of $N_p \times (1 + N_{\max\ neighbour})$ as shown in Figure 5.8

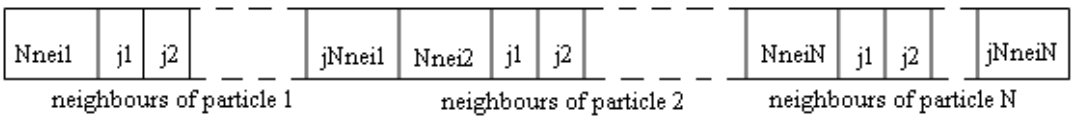


Figure 5.8: Neighbouring particle list in Verlet list algorithm

And another array called *point* with dimension of N_p is needed to point to the first neighbour for each particle (Ellero, et al.). For example: the number of neighbouring particles of particle 2 is Nnei2. In the point array,

$point[2] = neighbourlist[\text{Nnei1} + 1]$=j1

This means that the first neighbouring particle of particle 2 is the $[\text{Nnei1} + 1]$th particle stored in the *neighbourlist* array.

Summary

With this neighbouring particle list the calculation of distances between any two particles can be avoided for N_m times so the total computation time can be reduced. And for the algorithm includes several correction steps within one time step the neighbouring list will help reduce computation time significantly. However, the neighbouring particle searching algorithm is still $O(N^2)$.

5.2.3 Combing Cell-Linked List and Verlet List algorithm

In this case, the problem domain is discretized into regular cells. The size of the cells can be larger or smaller than the cut-off distance. The particles are allocated in appropriate cells according to their coordinates. A neighbouring particle list of each particle is built by comparing the distances with the cut-off distance. Only the particles in the same cell and adjacent cells are assessed for the Verlet list construction. This combination requires one array with size of $N_{cx} \times N_{cy} \times N_{cz}$ to store the index of cells, an array with size of $N_{cx} \times N_{cy} \times N_{cz}$ to store the numbers of particle inside each cell; an array of size of $N_{cx} \times N_{cy} \times N_{cz} \times N_{pc}$ to store the actual particles inside of each cell (N_{pc} is the maximum number of particles inside one cell); an array of size N_p to store the number of neighbouring particles for each particle and an array with size of $N_p \times N_{\max neighbour}$ to store the neighbouring list for each particle. It seems that more memory space is needed but for complex solution processing it can save significant computational time.

The efficiency can be further improved if only the neighbour cells with higher index are considered. As shown in Figure 5.9, sweeping through the grid along the x-direction, around each cell, only the North-west, North, North-east and East neighbouring cells are checked (Gesteria, et al. 2010). Taking cell (4,4) in column 4, row 4 for example, the target cells are (3,5), (4,5), (5,5) and (5,4). The rest of the neighbouring cells have been considered previously in the process (e.g. the neighbouring checking between cell (4,4) and (3,4) having been previously accounted for when cell (3,4) was the centre cell) .

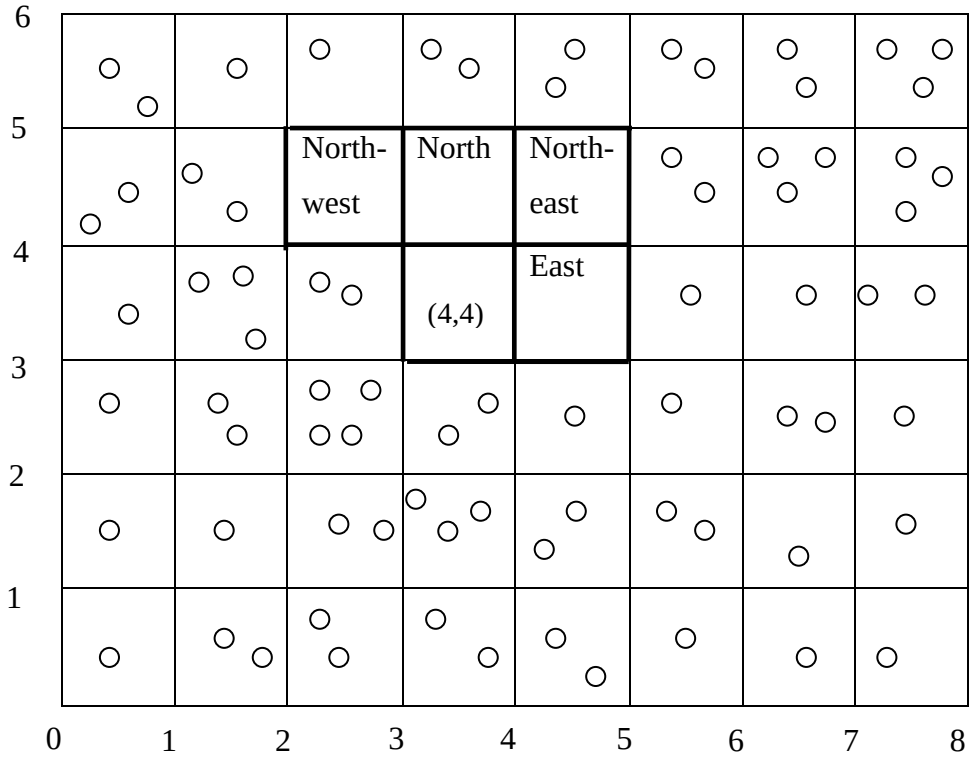


Figure 5.9: sweeping through grid cells considering pair-wise relation among neighbouring cells.

(*Starting from the lower left corner, particles inside the centre cell (4,4) interact with adjacent cells only in North-west, North, North-east and East directions. The interactions with the rest of cells in West, South, South-west and South-east were previously computed using reverse interactions*)

Hence, distance needs to be computed only $5 \times N_{pc} \times N_p$ times for all the particles rather than $N_p \times N_p$ times.

Alternatively, symmetrical characteristic is considered for particle pairs directly rather than the cells as shown in Figure 5.10. Taking particle *i* for example, all the particles in the nine contiguous cells are possible to be its neighbouring particles. But only the particles with a higher index will be checked.

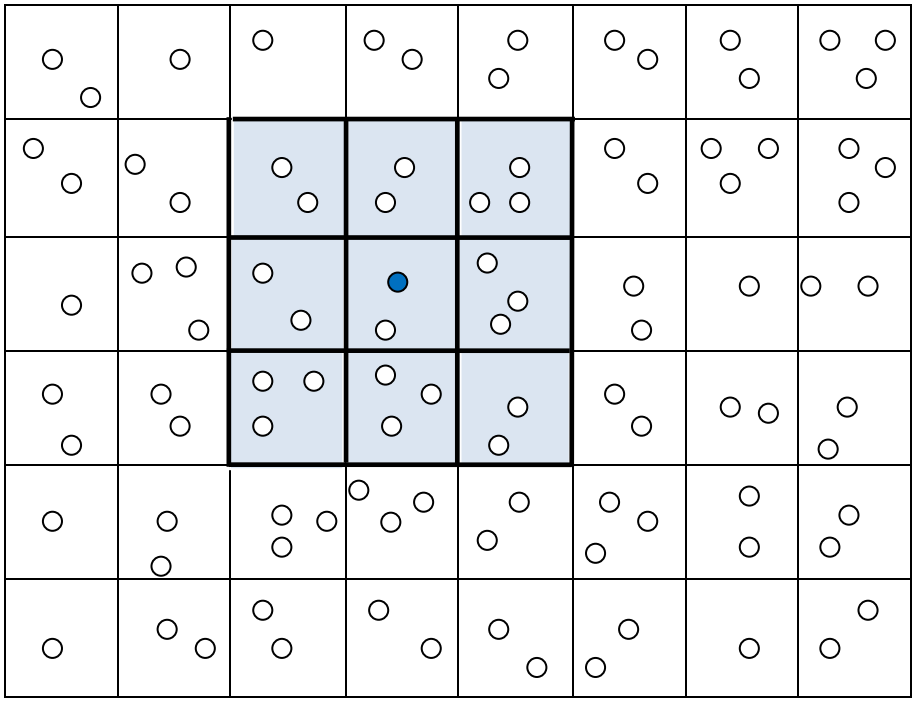


Figure 5.10: using pair-wise relation for particles directly instead of adjacent cells

The related pseudo-code can be written as

if $j > i$

do $r = \sqrt{(x[i]\text{-}x[j])^2 + (y[i]\text{-}y[j])^2}$

if $r < r_{cut}$

j is the neighbour particle of i;

i is the neighbour particle of j.

Namely, the particles with a smaller index than i will not be included in the computation loop. If the distance between i and j is smaller than the cut-off distance, then j is recorded in the neighbouring list of i, and at the same time, i is recorded in the neighbouring list of j. So when j is the centre particle, its neighbouring particles which have a smaller index such as particle i are already stored in its neighbouring list and so they will not be checked in the searching loop.

In this case when the pair-wise relation of the particles is considered, only $4.5 \times N_{pc} \times N_p$ calculations need to be performed for all the particles. This is better than considering the pair-wise relation of the cells.

5.3 Time stepping algorithm

The same time step size is used for both fluids and solids. When ISPH method is applied the time stepping algorithm for fluid is divided into two steps as discussed in section 4.1. In the first step an intermediate velocity for fluid is computed without considering the effect of pressure. The values obtained from the first step will be adjusted in the second step through the effect of pressure. Finally the position of each particle would be updated according to the new velocities.

After the properties of the fluid are updated, the external force acting on the solid can be obtained by the summation of fluid force on the solid particles submerged in the water. So the velocity and new position of the solid can be calculated. The Euler method can be applied to update the physical properties of the solid as the time step size used for SPH method is normally very small. The Euler method is a very simple way to integrate a general function $A(t)$ as follows:

$$A(t+\delta t) = A(t) + \delta t \dot{A} \tag{5.3.1}$$

Since pressure is calculated implicitly in the ISPH method but other properties are calculated explicitly, the size of the time step must be controlled in order to generate stable and accurate results. The following Courant-Friedrichs-Levy (CFL) condition must be satisfied (Shao & Lo 2003)

$$\Delta t \le 0.1 \frac{r}{\mathbf{v}_{\max}} \tag{5.3.2}$$

where r is the initial particle spacing and $\mathbf{v}_{\max}$ is the maximum particle velocity in the computation. The factor 0.1 is introduced to ensure that the particle moves only a fraction (in this case 0.1) of the particle spacing per time step. When viscous diffusion is considered another constraint on time step size needs to be satisfied (Cummins & Rudman 1999)

$$\Delta t \le 0.125 \frac{h^2}{\mu / \rho} \tag{5.3.3}$$

The allowable time-step size should satisfy both of the above criteria.

Chapter 6 Studies of the effect of modelling parameters in SPH

As in any numerical methods, the performance of SPH in terms of accuracy, stability and computational time can be influenced by a number of parameters such as the choice of kernel function, smoothing length, time step size, and number of particles and so on. In this chapter, investigation of the effect of kernel functions, number of particles and smoothing length is carried out in a one dimensional case by applying SPH approximation to represent a number of common functions. A dam breaking case is used as a two dimensional example to study the effects of kernel functions, time step sizes and particle numbers.

6.1 The effect of different kernels

Kernel function is one of the key components in SPH method. This section focuses on the study of the effect of different kernel functions with different smoothing lengths and particle numbers. Nine popularly used kernel functions (Liu, et al. 2003a) are considered.

Kernel functions investigated:
1). Quadratic (Hicks & Liebrock 2000)

$$W(r,h)=\frac{3}{4h}\left(1-\left(\frac{r}{h}\right)^2\right), \qquad 0\le\frac{r}{h}\le 1 \tag{6.1.1}$$

This quadratic smoothing function was used in the grid free finite integration method. The main advantage of this kernel function is the simplicity and easy for computation whereas the drawback is that the first derivative is not zero on the boundary of the support domain, which means that it does not have compact support for its first derivative.

2). Quartic (Lucy 1977)

$$W(r,h)=\frac{5}{4h}\left(1-6\left(\frac{r}{h}\right)^2+8\left(\frac{r}{h}\right)^3-3\left(\frac{r}{h}\right)^4\right),\ 0\le\frac{r}{h}\le 1 \qquad (6.1.2)$$

This quartic smoothing function and its first two derivatives satisfies the compact support condition.

3). Johnson's quadratic

$$W(r,h)=\frac{1}{h}\left(\frac{3}{16}\left(\frac{r}{h}\right)^2-\frac{3}{4}\frac{r}{h}+\frac{3}{4}\right),\qquad 0\le\frac{r}{h}\le 2 \qquad (6.1.3)$$

The speciality of this kernel is that the first derivative increases as the particles move closer and it decreases as they move apart. This is advantageous for adjusting the position of particles to maintain the stability. However, the derivative of this kernel function is not smooth at $r=0$.

4). Gaussian (Gingold & Monaghan 1977)

$$W(r,h)=\frac{1}{\sqrt{\pi}h}e^{-\left(\frac{r}{h}\right)^2},\qquad 0\le\frac{r}{h}\le 2 \qquad (6.1.4)$$

Gaussian kernel was used to simulate the non-spherical stars originally. It is sufficiently smooth even for the second order derivative. However, it is not really compact as it never goes to zero theoretically. This can result in a large support domain with an inclusion of many particles for the particle approximation.

5). Super-Gaussian (Monaghan & Poinracic 1985a)

$$W(r,h)=\frac{1}{\sqrt{\pi}h}\left(\frac{3}{2}-\left(\frac{r}{h}\right)^2\right)e^{-\left(\frac{r}{h}\right)^2},\qquad 0\le\frac{r}{h}\le 2 \qquad (6.1.5)$$

This is one of the higher order smoothing functions that are devised from lower order forms. Its main disadvantage is that the kernel is negative in some region of its support domain. This may lead to unphysical results for hydrodynamics problems.

6). Cubic-spline (Monaghan & Poinracic 1985a)

$$W(r,h)=\begin{cases}\frac{1}{h}\left(\frac{2}{3}-\left(\frac{r}{h}\right)^2+\frac{1}{2}\left(\frac{r}{h}\right)^3\right) & 0<r<h \\ \frac{1}{6h}\left(2-\frac{r}{h}\right)^3 & h\le r<2h \\ 0 & r\ge 2h \end{cases} \qquad (6.1.6)$$

The cubic spline function is the most widely used smoothing functions since it resembles a Gaussian function while having a compact support. However, the second derivative of the cubic spline is a piecewise linear function, the stability properties can be inferior to those of smoother kernels.

7). Quartic-spline

$$W(r,h)=\begin{cases}\frac{1}{24h}\left(\left(\frac{r}{h}+2.5\right)^4-\left(5\left(\frac{r}{h}+1.5\right)^4+10\left(\frac{r}{h}+0.5\right)^4\right)\right) & 0<r<0.5h \\ \frac{1}{24h}\left(\left(2.5-\frac{r}{h}\right)^4-5\left(1.5-\frac{r}{h}\right)^4\right) & 0.5h\le r<1.5h \\ \frac{1}{24h}\left(2.5-\frac{r}{h}\right)^4 & 1.5h\le \frac{r}{h}<2.5h \\ 0 & r\ge 2.5h \end{cases} \qquad (6.1.7)$$

Higher order splines were introduced because they are better approximation of the Gaussian smoothing kernel and more stable.

8). Quintic spline

$$W(r,h)=\begin{cases}\dfrac{1}{120h}\left(\left(3-\dfrac{r}{h}\right)^5-6\left(2-\dfrac{r}{h}\right)^5+15\left(1-\dfrac{r}{h}\right)^5\right) & 0<r<h\\ \dfrac{1}{120h}\left(\left(3-\dfrac{r}{h}\right)^5-6\left(2-\dfrac{r}{h}\right)^5\right) & h\le r<2h\\ \dfrac{1}{120h}\left(3-\dfrac{r}{h}\right)^5 & 2h\le \dfrac{r}{h}<3h\\ 0 & r\ge 3h\end{cases} \tag{6.1.8}$$

9). New-quartic (Liu, et al. 2003a)

$$W(r,h)=\frac{1}{h}\left(\frac{2}{3}-\frac{9}{8}\left(\frac{r}{h}\right)^2+\frac{19}{24}\left(\frac{r}{h}\right)^3-\frac{5}{32}\left(\frac{r}{h}\right)^4\right),\qquad 0\le\frac{r}{h}\le 2 \tag{6.1.9}$$

This quadratic smoothing function satisfies the compact support for the first derivative and it has a smoother second derivative than the piecewise linear second derivative of the cubic function, and therefore the stability properties should be superior to those of the cubic function. However, the second derivative is not monotonic function of *r*. This may lead to an incorrect approximation.

These kernel functions are used to approximate 5 common functions in one dimensional case.

Functions approximated

Function 1: $f=x$; Function 2 : $f=x^3$; Function 3: $f=e^{-x}$;

Function 4: $f=\sin(x)$; Function 5: $f=\tan(x)$

6.1.1Accuracy analysis

Approximation in SPH involves two steps. First, a function is approximated in integral form and then transformed into particle approximation. These two

approximations are investigated separately. In each approximation, the effect of different smoothing length and different particle numbers are considered separately.

Integral approximation

Similar to many other numerical methods, discretization is required. As shown in Figure 6.1, the problem domain is divided uniformly into a collection of points with spacing of δr, and then the smoothing domain of each point is further divided into a set of points with spacing of $\delta r'$.

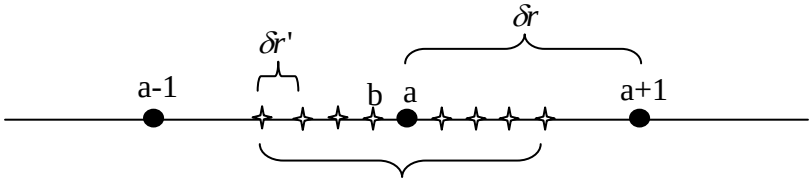


Figure 6.1: Discretization in integral approximation

the kernel approximation is then formulated as

$$\begin{aligned}\langle f(x_a)\rangle &= \int_{\Omega} f(x_b) W(x_a - x_b)\,dx' \\ &= \sum_{b=1}^{N} f(x_b) W_{ab}\,\delta x'\end{aligned} \tag{6.1.10}$$

Here, $\delta x'$ is the spacing of points within the smoothing domain, i.e. $\delta x' = \delta r'$. The analytical result is calculated and compared with the integral approximation results, and then the error is obtained.

Particle approximation

In particle approximation $\delta x'$ will be replaced by particle volume which is related to material properties. In order to keep consistent with the integral approximation, the volume of a particle should satisfy the following condition

$$\delta x' = \frac{m_b}{\rho_b} \tag{6.1.11}$$

So the density can be expressed as

$$\rho = \frac{m_b}{\delta x'} \tag{6.1.12}$$

The mass of the system is assumed to be a unit as no specific material is considered. Since a uniform distribution of mass is preferred the discretization in particle approximation is slightly different from integral approximation. The problem domain is just uniformly divided into a number of particles as shown in Figure 6.2, i.e. in this case $\delta r = \delta r'$.

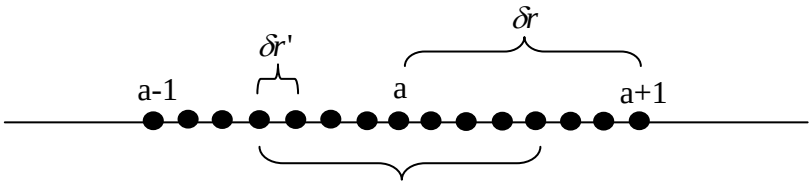


Figure 6.2: Discretization in particle approximation

Applying integral approximation to Equation (6.1.12)

$$\langle \rho \rangle = \sum_{b=1}^{N} \frac{m_b}{\delta x'} W_{ab} \delta x' = \sum_{b=1}^{N} m_b W_{ab} \tag{6.1.13}$$

Hence, the particle approximation at each particle is expressed as

$$\begin{aligned} \langle f(x_a) \rangle &= \sum_{b=1}^{N} f(x_b) W_{ab} \delta v \\ &= \sum_{b=1}^{N} f(x_b) W_{ab} \frac{m_b}{\rho_b} \\ &= \sum_{b=1}^{N} f(x_b) W_{ab} \left(\frac{1}{\sum_{b=1}^{N} W_{ab}} \right) \end{aligned} \tag{6.1.14}$$

The main difference between Equation (6.1.14) and (6.1.10) is the volume (spacing) representation. The difference of discretization between integral approximation and particle approximation is the results of the introduction of material property.

Investigation of individual kernel functions

It is obvious that smoothing length and particle numbers are the two key factors affecting the accuracy of a kernel approximation. Hence, the accuracy of each kernel function is tested with different smoothing length and different particle numbers by checking the errors from the two approximations. The values of error are calculated using Root Mean Square (RMS) method

$$\varepsilon = \sqrt{\frac{1}{N}\sum_{\mathrm{b}=1}^{N}\varepsilon_{\mathrm{b}}^{2}} \tag{6.1.15}$$

where ε is the RMS error, ε_b is the error between SPH result and analytical results at the b th particle; N is the number of particles used for error analysis.

For more detailed analysis, the error is assumed to be related to the number of neighbouring particles and the value of smoothing length as $\varepsilon \propto AN^{\alpha}$ and $\varepsilon = Bh^{\beta}$ respectively. Where α and β are variables to be determined, N is the particle numbers, h is smoothing length and A, B are coefficients. Based on these assumptions, when logarithmic scales are used, the relationship between ε and N will appear as a straight line and so is the case for ε and h.

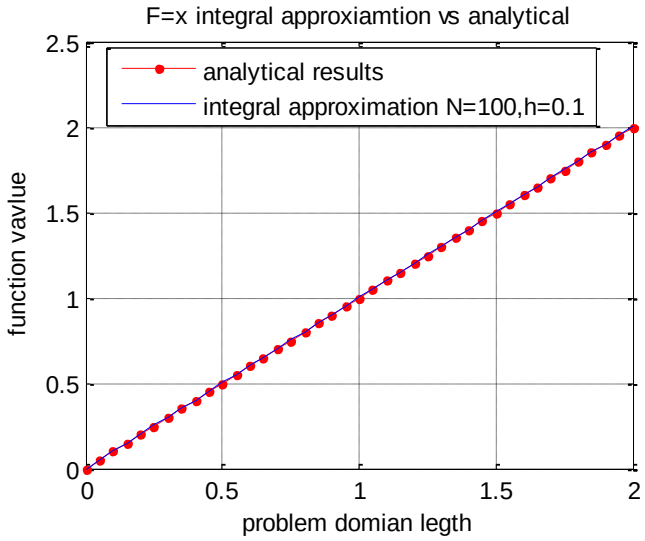
The gradient between RMS error and smoothing length in a log-log plot is used to determine how smoothing length affects the accuracy (shown in Figure 6.4 and 6.7) similarly, the gradient of the error against the particle numbers is used to find out how to improve the accuracy with particle numbers (shown in Figure 6.5 and 6.8).

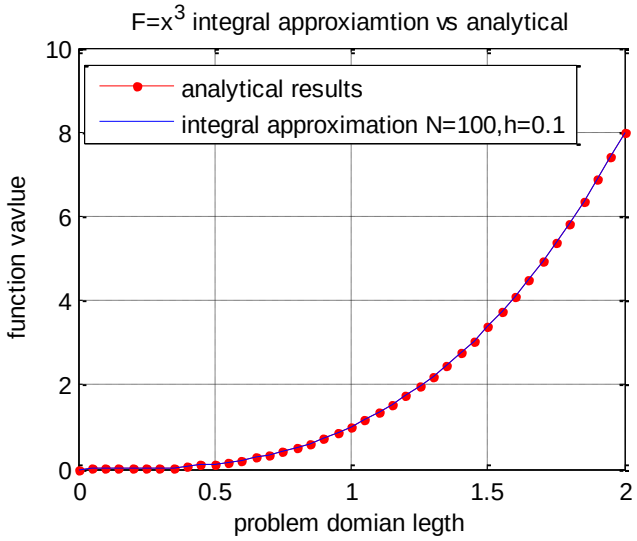
The appropriate smoothing length can be determined based on the diagrams of RMS error against smoothing length; an error less than 1% is chosen to be the criteria and so for particle numbers. The better kernel function should give lower error.

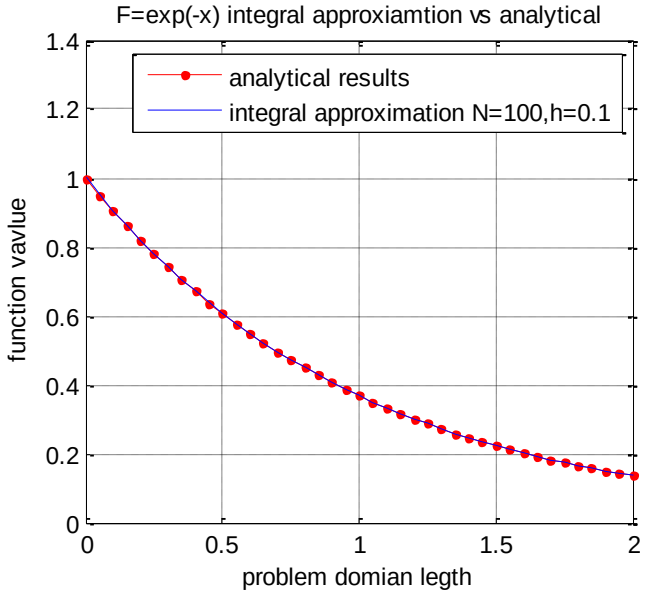
6.1.2 Results and discussions

1). Integral approximation

Results obtained with SPH integral approximation and analytical results are shown below in Figure 6.3







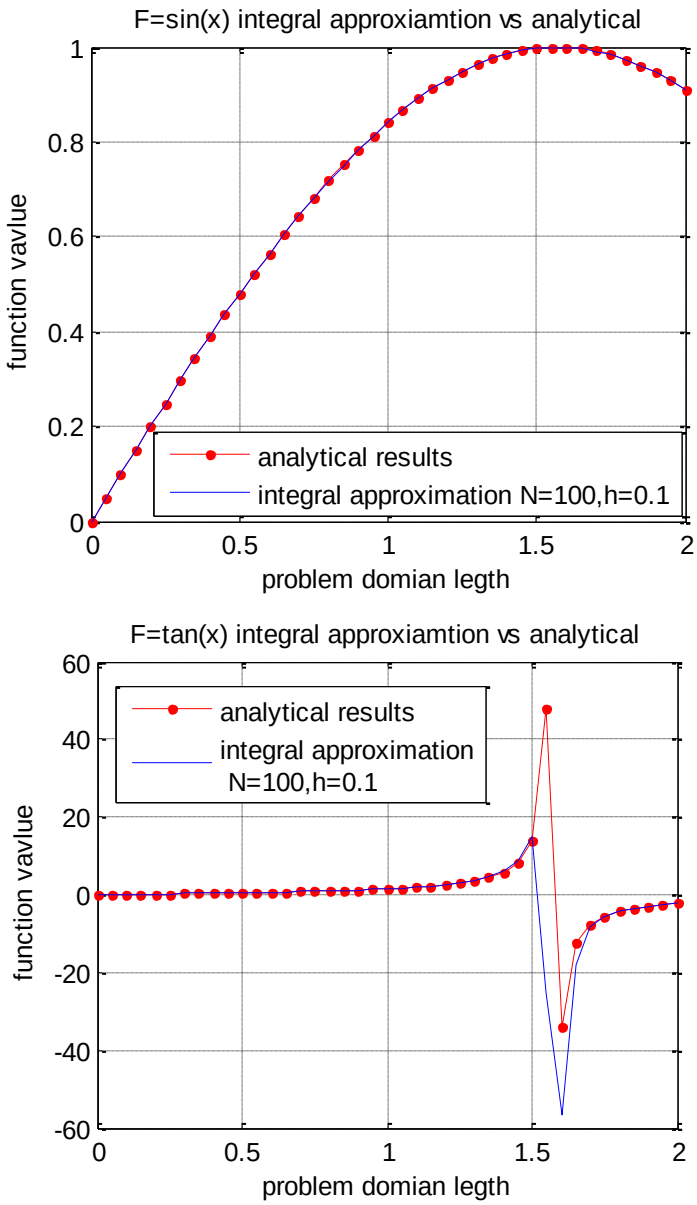
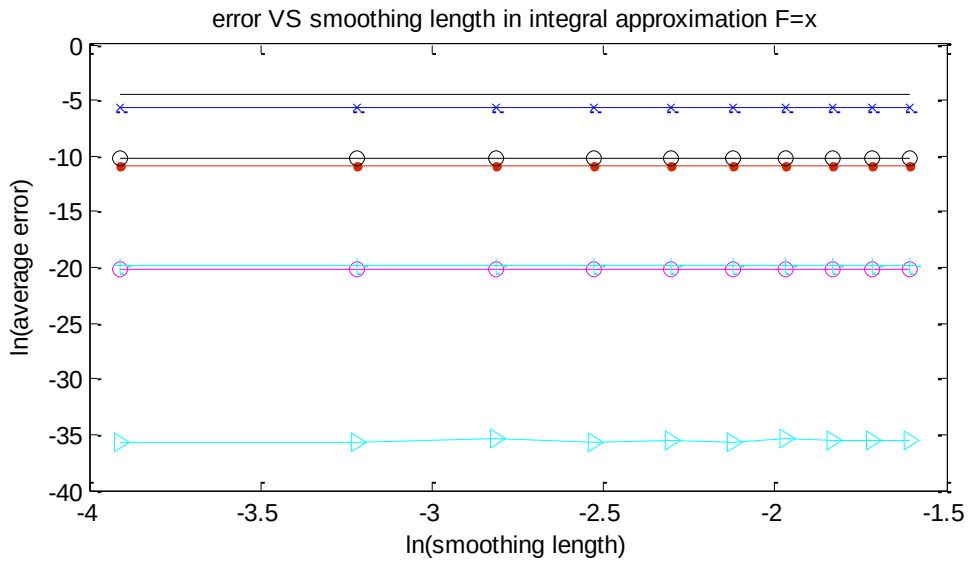
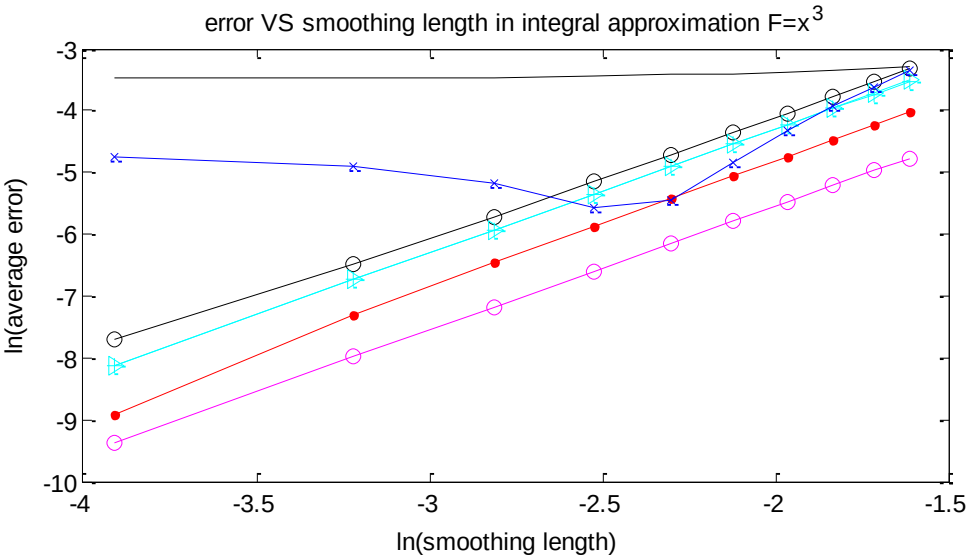


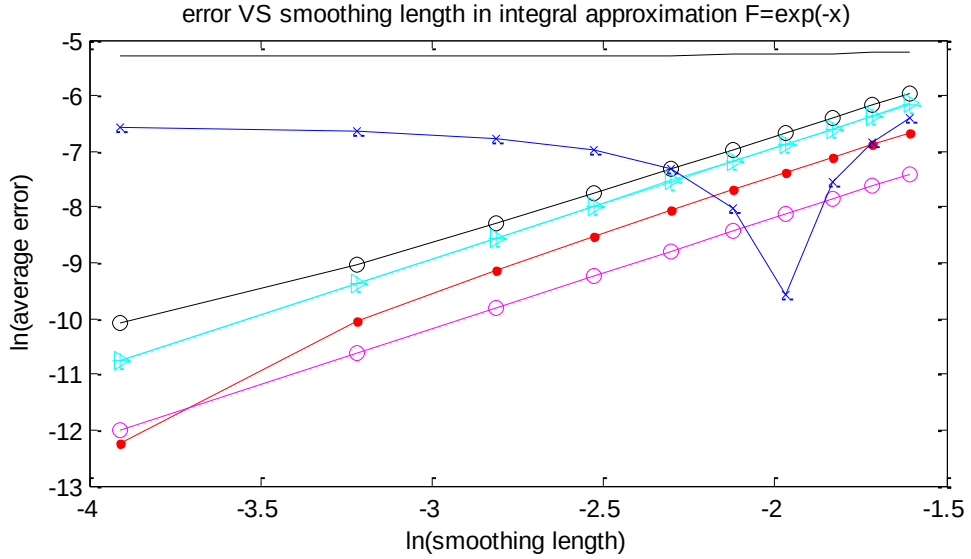
Figure 6.3: Integral approximation results for various functions: $F = x$, $F = x^3$, $F = e^{-x}$, $F = \sin(x)$ and $F = \tan(x)$

In Figure 6.3 it shows that the results obtained from integral approximation agree well with the analytical data except for function $F = \tan(x)$ due to the singularity of the function. In this case the problem domain should be divided into two sections to avoid incorrect computation through the singular point.

The influence of smoothing length and the number of neighbouring particles on errors in integral approximation is studied next and the results are shown below in Figure 6.4 and 6.5 respectively.


error VS smoothing length in integral approximation F=x
ln(average error)
ln(smoothing length)
0
-5
-10
-15
-20
-25
-30
-35
-40
-4
-3.5
-3
-2.5
-2
-1.5


error VS smoothing length in integral approximation F=x3
ln(average error)
ln(smoothing length)
-3
-4
-5
-6
-7
-8
-9
-10
-4
-3.5
-3
-2.5
-2
-1.5


error VS smoothing length in integral approximation F=exp(-x)
ln(average error)
ln(smoothing length)
-5
-6
-7
-8
-9
-10
-11
-12
-13
-4
-3.5
-3
-2.5
-2
-1.5

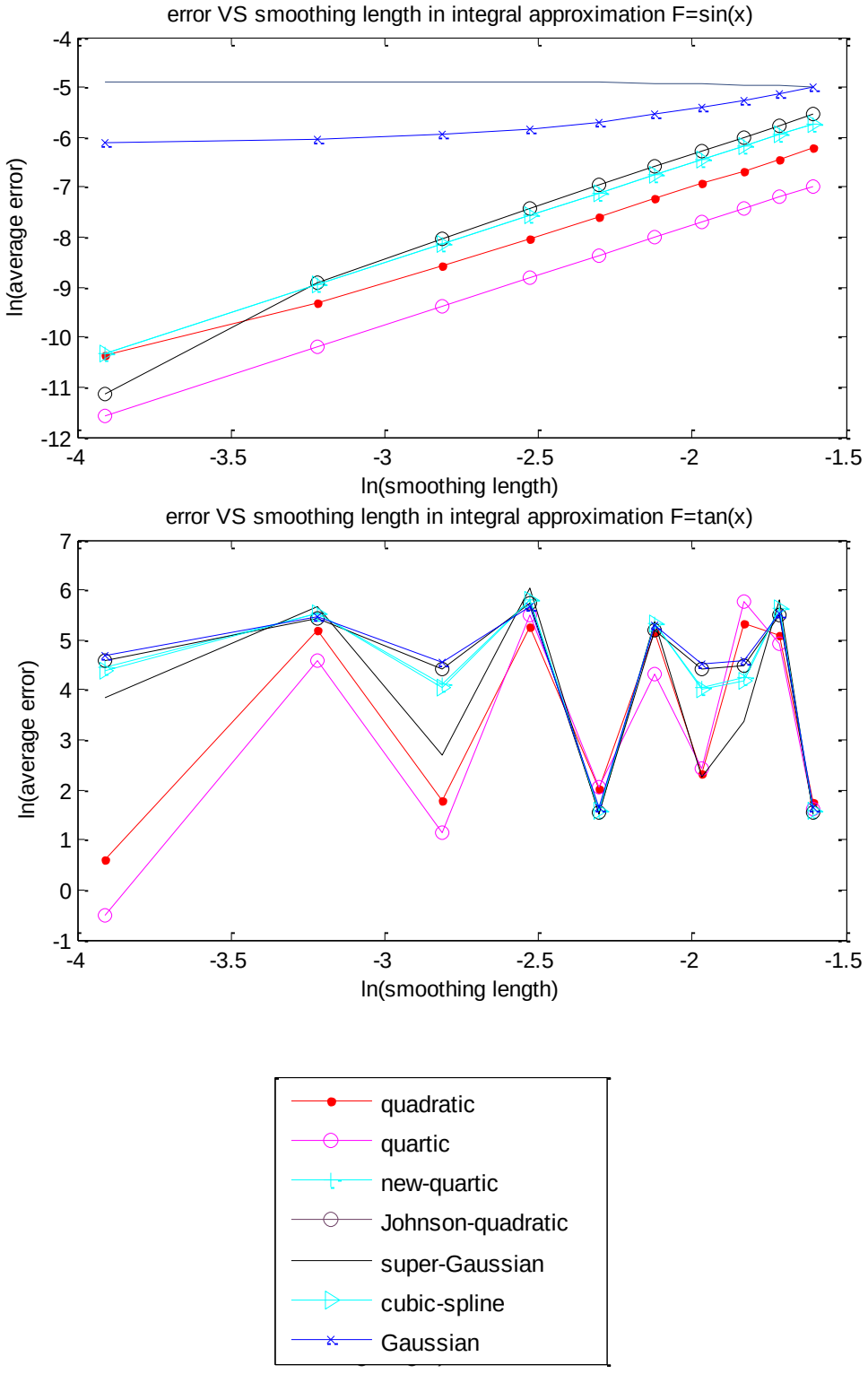
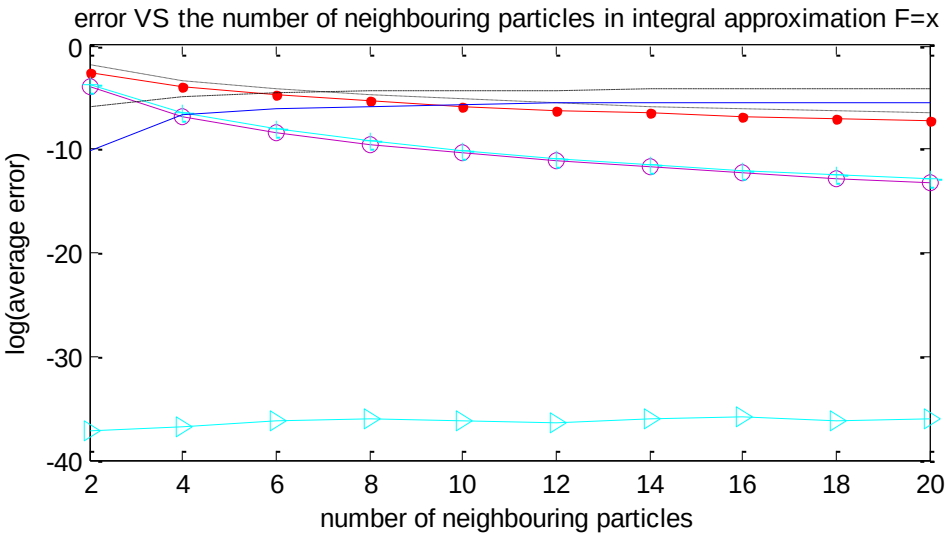
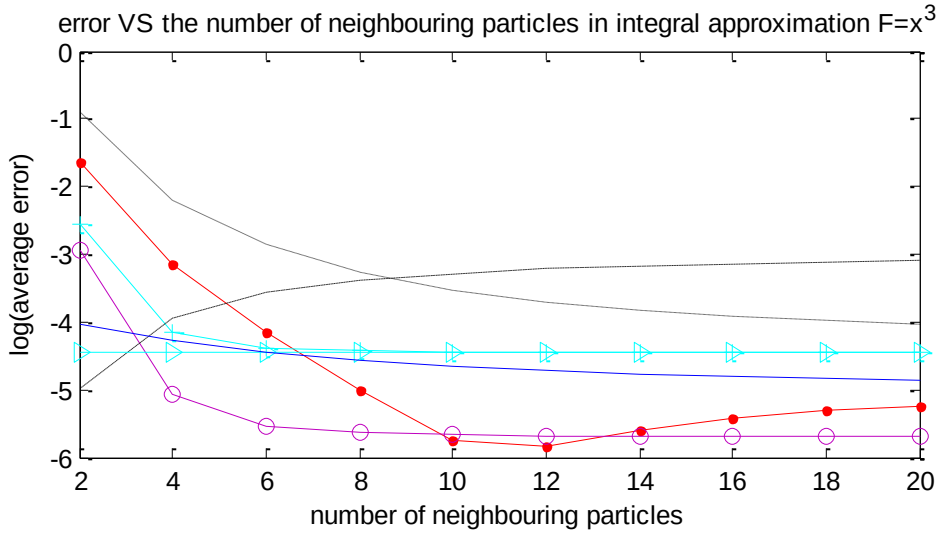


Figure 6.4: Results obtained from integral approximation with changing smoothing length in the problem domain (N=100). *The legend applies to the following figures of error analysis as well*

From Figure 6.4 it can be seen that in most cases decreasing the smoothing length can reduce the error. The function $f = \tan(x)$ is a special case. The error is high and it fluctuates. It is difficult to improve the results by decreasing the smoothing length. The reason for this is that this function is singular in the domain. Overall, the super-Gaussian kernel function provides smaller errors and the results are not sensitive to the change of smoothing length.





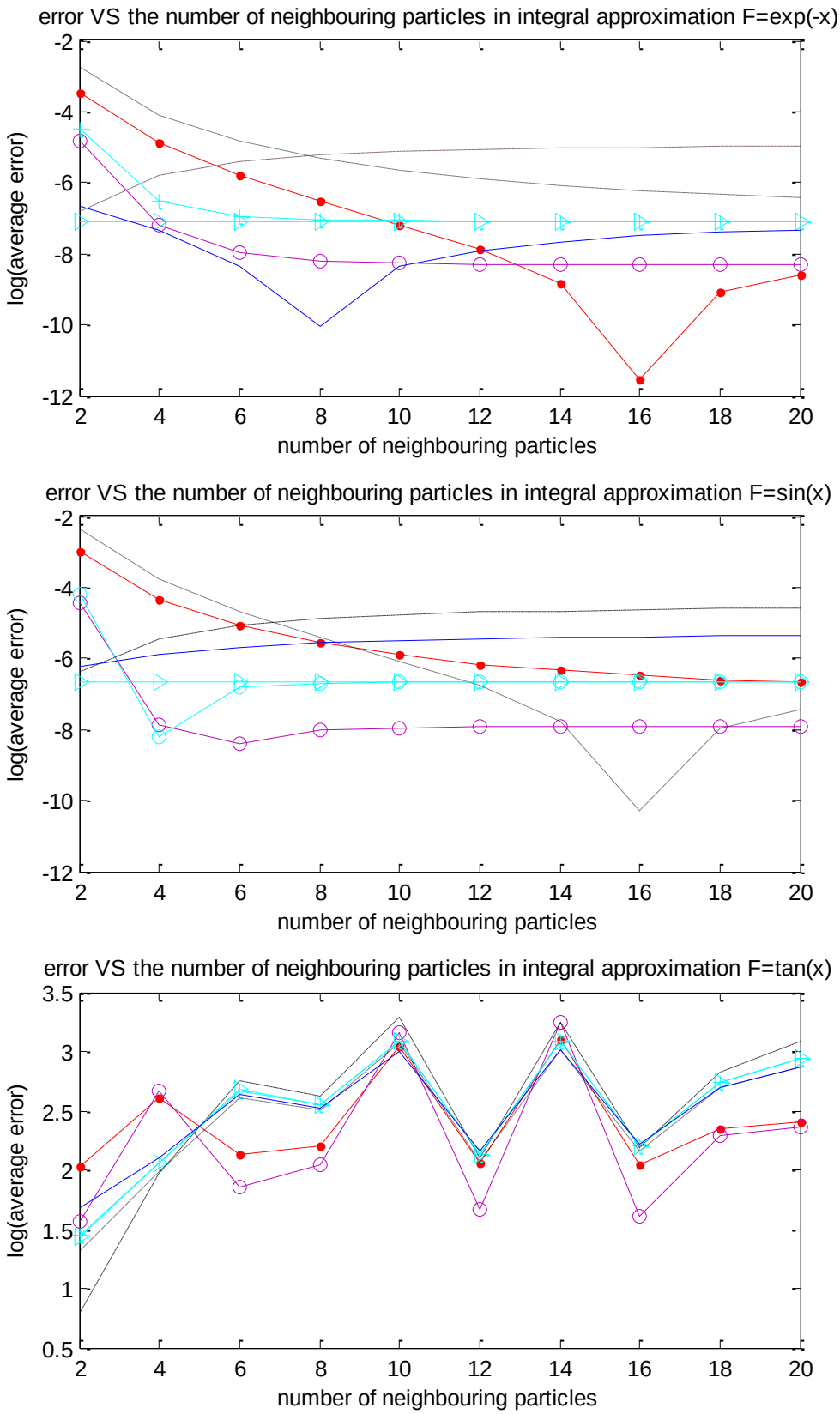
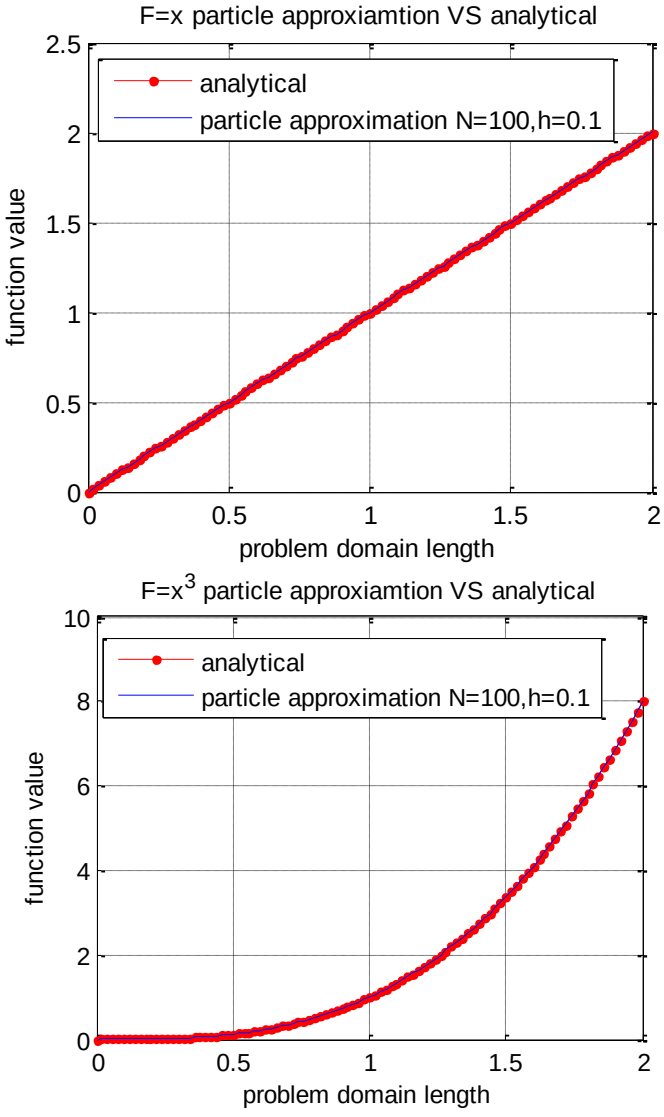


Figure 6.5: Results obtained from integral approximation with particle numbers increasing inside the smoothing length (h=0.1)

The same legend as shown in Figure 6.4 is used for all the figures on error analysis. From Figure 6.5 it is noted generally that to have 5 neighbouring particles can produce an approximation with an error under 5% for all kernel functions except for function $f = \tan(x)$. This means that the ratio of about 2 between the spacing and the smoothing domain radius (*kh*) can provide accurate results. Increasing the number of neighbouring particles can slightly improve the accuracy of the approximation for most kernels except cubic spline kernel. It seems that quartic kernel provides the smallest error in most cases and cubic spline kernel is not sensitive to the decreasing smoothing length.

2). Particle approximation

Results obtained with SPH particle approximations and analytical results are shown below in Figure 6.6.



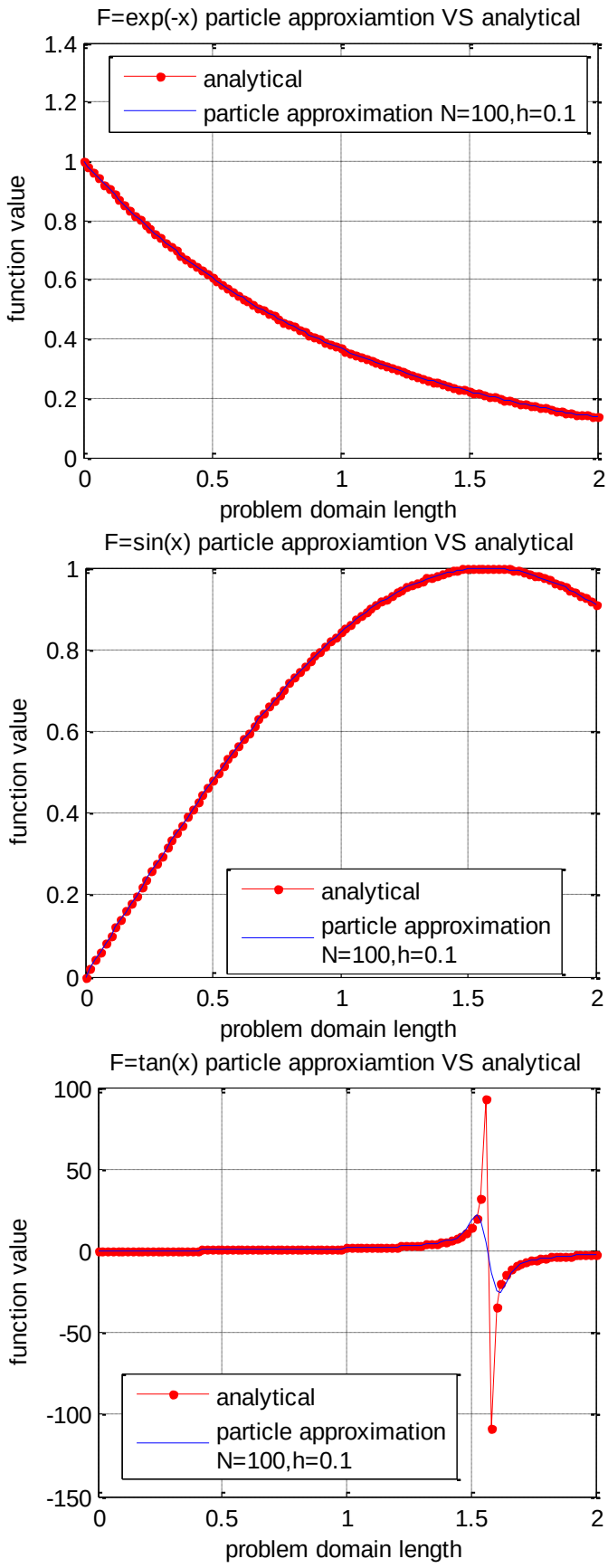
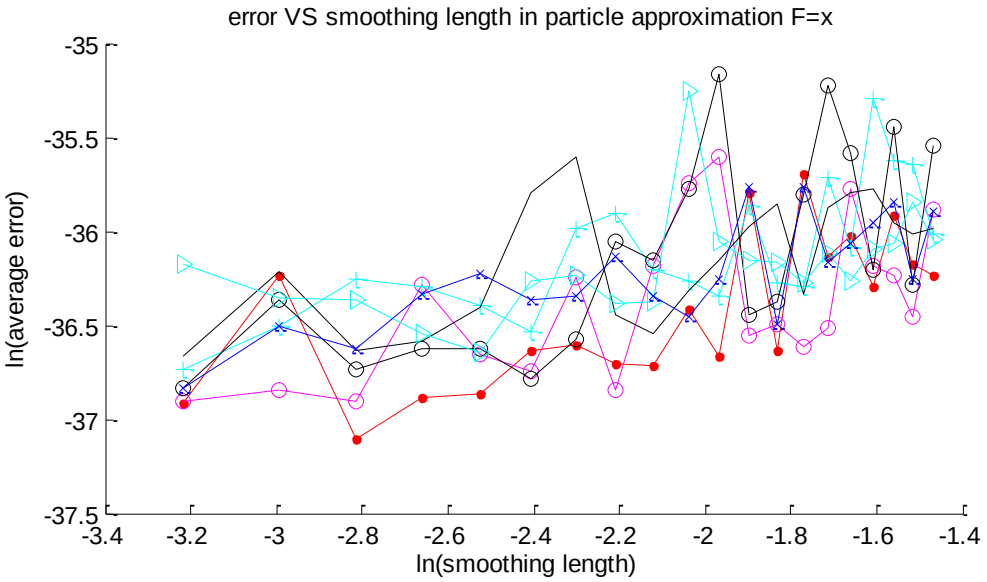
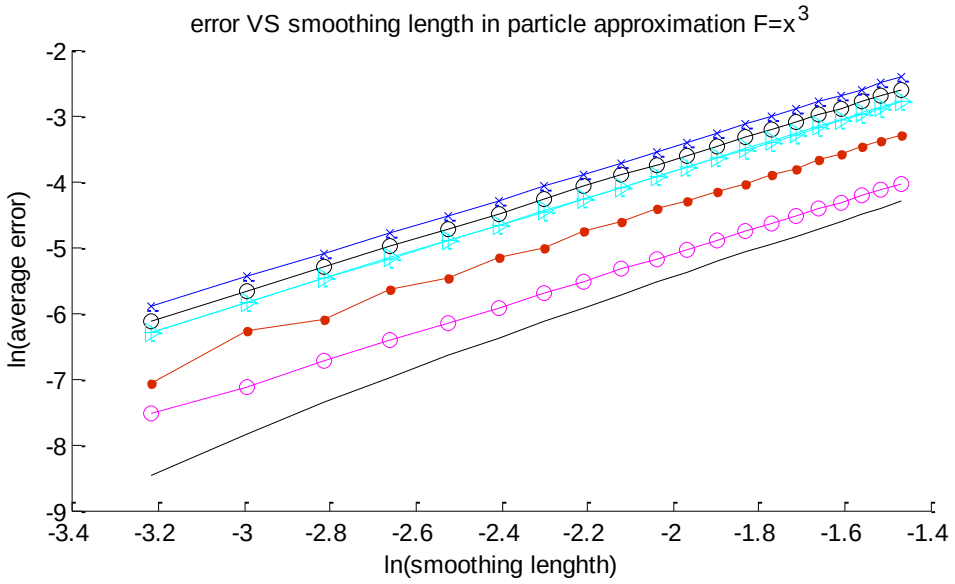


Figure 6.6: particle approximation results for various functions: $F = x$, $F = x^3$, $F = e^{-x}$, $F = \sin(x)$ and $F = \tan(x)$

The results obtained from particle approximation also agree with the analytical data except for function $F = \tan(x)$. Influence of smoothing length and the number of particles on errors in particle approximation is studied and the results are shown below in Figure 6.7 and 6.8 respectively.





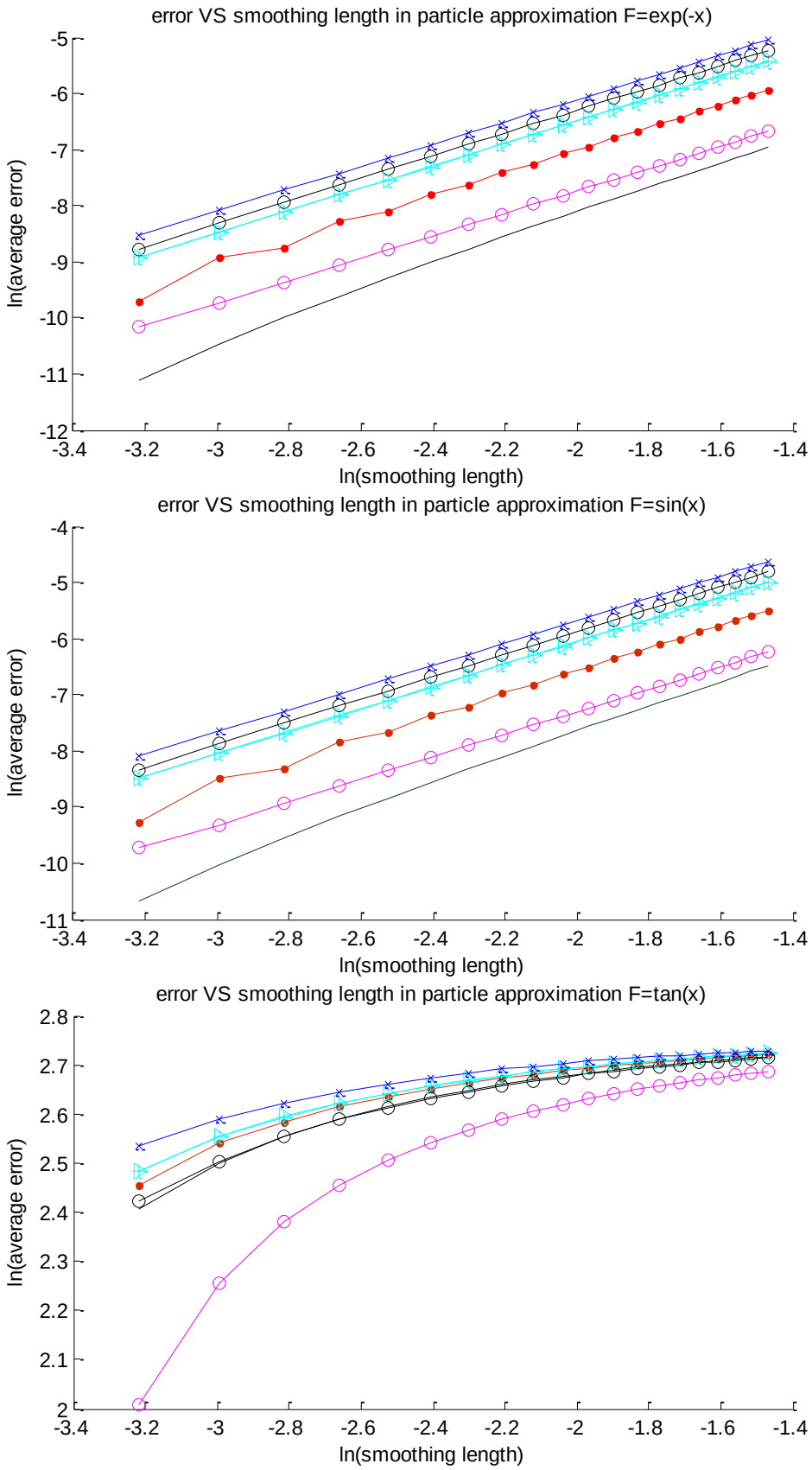
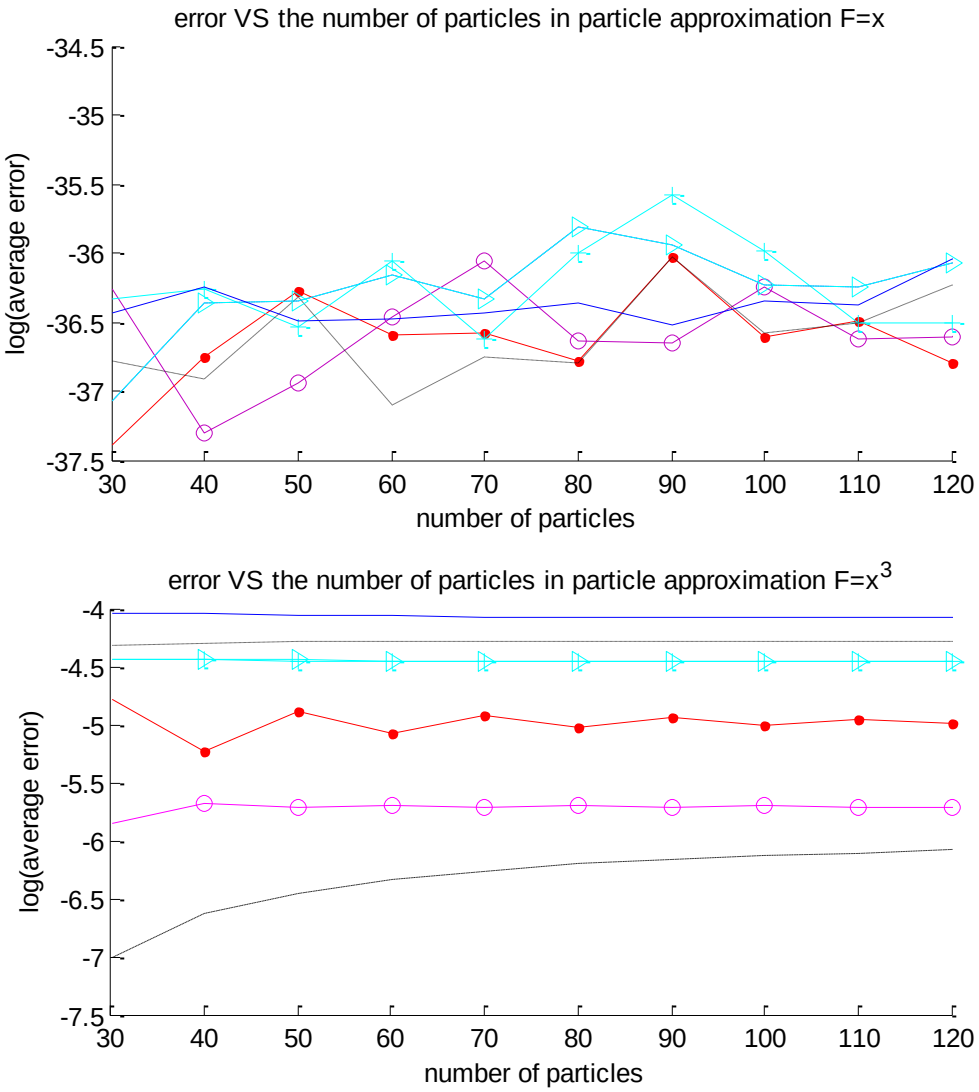


Figure 6.7: Results obtained from particle approximation with changing smoothing length (N=100)

From Figure 6.7, it is clear that a smaller smoothing length provides smaller error. In the case of linear function approximation, the error is very small though the curves are not showing a clear trend. For liner function $F = x$, the results are normally very accurate even with a large smoothing length. Therefore, decreasing the smoothing length is not meaningful in this case. It seems that the supper Gaussian gives the best accuracy in this particle approximation and quartic is the second best.



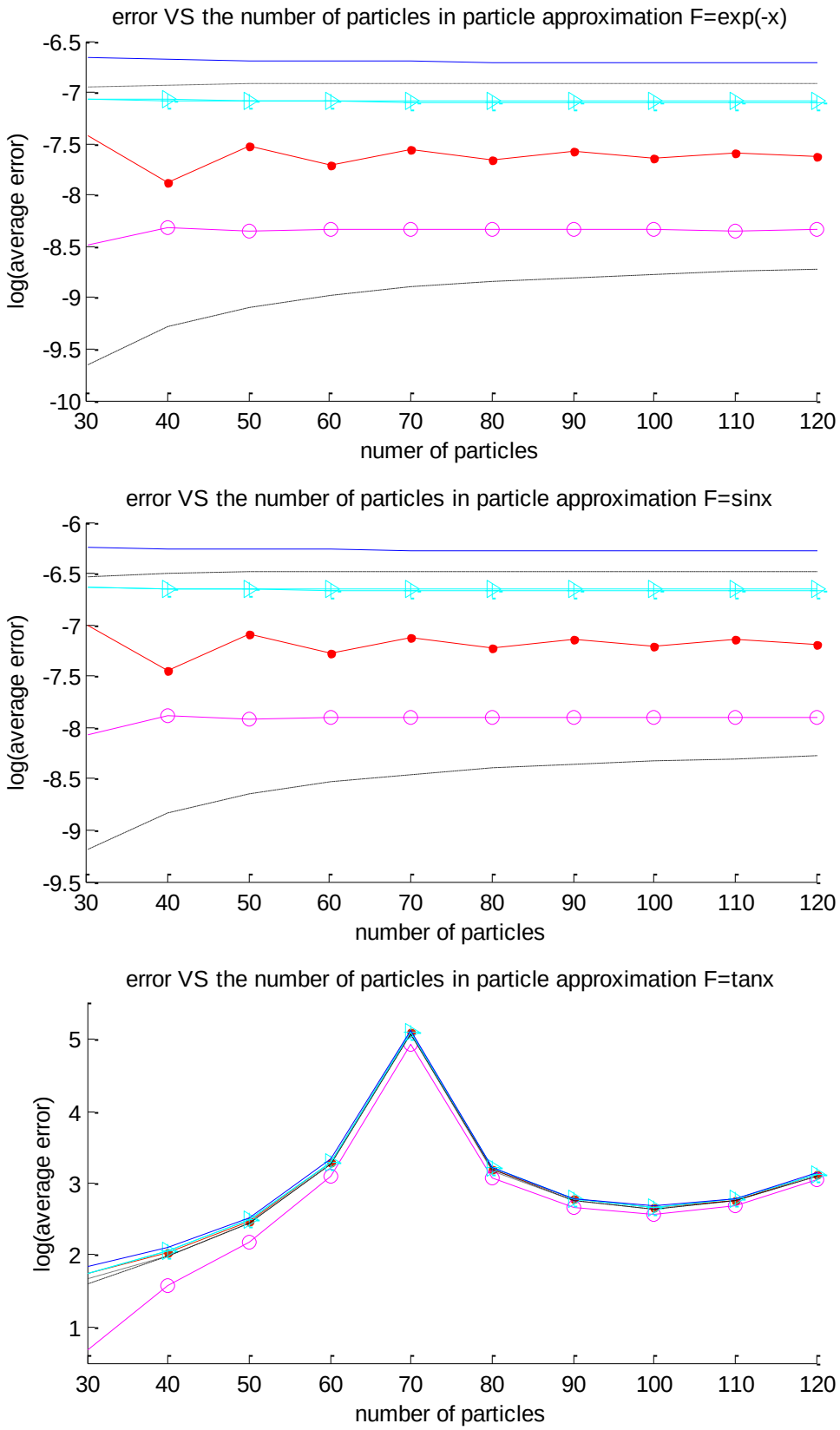


Figure 6.8: Results obtained from particle approximation changing particle numbers (h=0.1) (In this method, error obtained for $F = x$ is very small with all these kernel functions)

In Figure 6.8, for functions $F = x^3$, $F = \sin(x)$ and $F = e^{-x}$, the accuracy can be improved by increasing the particle numbers same as for integral approximation. For function $F = x$ the error is really small although the curve fluctuates with decreasing smoothing length. The supper-Gaussian kernel can provide the lowest error in most cases. For function $F = \tan(x)$, the error is quite large with all the kernels.

Comparing integral approximation and particle approximation we can see that the error can be reduced with smaller smoothing length. The results can also be improved by increasing the number of particles but this is not as efficient as using smaller smoothing length. This indicates that a proper smoothing length is more important to obtain a correct approximation. For $F = \tan(x)$, as the value of the function approaches infinite at its singular point within the problem domain, it is difficult to obtain an accurate approximation. In this case, the problem domain can be divided into two sections and error can be assessed in each section.

In both two approximations, all the kernels produce similar results, the choice of kernel is not very important in most cases.

More detailed data from the error analysis for the kernel functions investigated are listed in tables 6.1 and 6.2. In the case where the effect of smoothing length is studied, the particle number is fixed to 100 and in the case where the effect of the number of particles is examined, the smoothing length is fixed to 0.1.

kernel function		Quadratic	Quartic	New-quartic	Johnson-quadratic	Super-Gaussian	Cubic-spline	Quadratic-spline	Quintic-spline	Gaussian
P-L	G	2.1677	1.9951	1.9990	2.0221	2.3737	2.0022	2.0028	2.0034	1.9906
P-N	G	0.0076	0.0434	0.0187	0.0377	0.0364	0.0124	0.0160	0.0148	0.011
I-L	G	2.4239	2.0003	2.0012	1.7935	0.0423	2.0013	2.0016	2.002	1.3694
I-N	G	0.4775	0.1954	0.1453	0.2036	0.1030	10^{-6}	0.0340	0.0882	0.1853

Table 6.1: Gradient of average error against particle spacing for function F=x

kernel function		Quadratic	Quartic	New-quartic	Johnson-quadratic	Super-Gaussian	Cubic-spline	Quadratic-spline	Quintic-spline	Gaussian
P-L	G	2.1666	1.9945	1.9970	2.0193	2.3844	2.000	2.000	2.000	1.9873
P-N	G	0.0051	0.0066	0.0066	0.0064	0.0215	0.0066	0.0066	0.0066	0.0083
I-L	G	2.1226	2.000	2.000	1.9003	0.0844	2.000	2.000	2.000	0.9576
I-N	G	0.2322	0.1526	0.1052	0.1738	0.1043	10^{-6}	0.0128	0.0531	0.0448

Table 6.2: Gradient of logarithm of average error against logarithm of particle spacing for function $F = \sin(x)$ *(Here, P-L and P-N stand for particle approximation with changing smoothing length and number of particles respectively, whereas I-L and I-N are for integral approximation with varying smoothing length and particle numbers inside the smoothing length respectively)*

The gradients of error against smoothing length are around 2 in both integral and particle approximation, which means that SPH approximation has a second order of accuracy with smoothing length. The influence of the number of particles on the errors is more noticeable in integral approximation than in particle approximation. This indicates that increasing the neighbouring particles is more efficient than increasing the total number of particles; the number of neighbouring particles determines the accuracy.

To summarise, integral approximation does not differ from particle approximation in most cases which means that the particle approximation is consistent with integral approximation. Different kernel functions give similar approximations. Especially for new quartic, cubic spline, quartic spline and quintic spline functions, they provide close results in most cases. It seems that the quartic kernel function shows the best performance generally. SPH approximation has a second order of accuracy with smoothing length. Decreasing the smoothing length or increasing the number of neighbouring particles is normally useful to improve the accuracy.

6.2 Investigation of various factors with WCSPH

Dam breaking problem is a classic benchmark problem for assessment of fluid simulations (Monaghan 1994). The model is shown in Figure 6.9. The water column is 0.09m (L=0.09m) by 0.18m (H=0.18m). At the starting moment of computation, the right side wall is removed instantaneously and the water column suddenly collapses as a result of gravity effect. The particles are assumed in hydrostatic state at the initial time instant. No-slip boundary condition is applied.

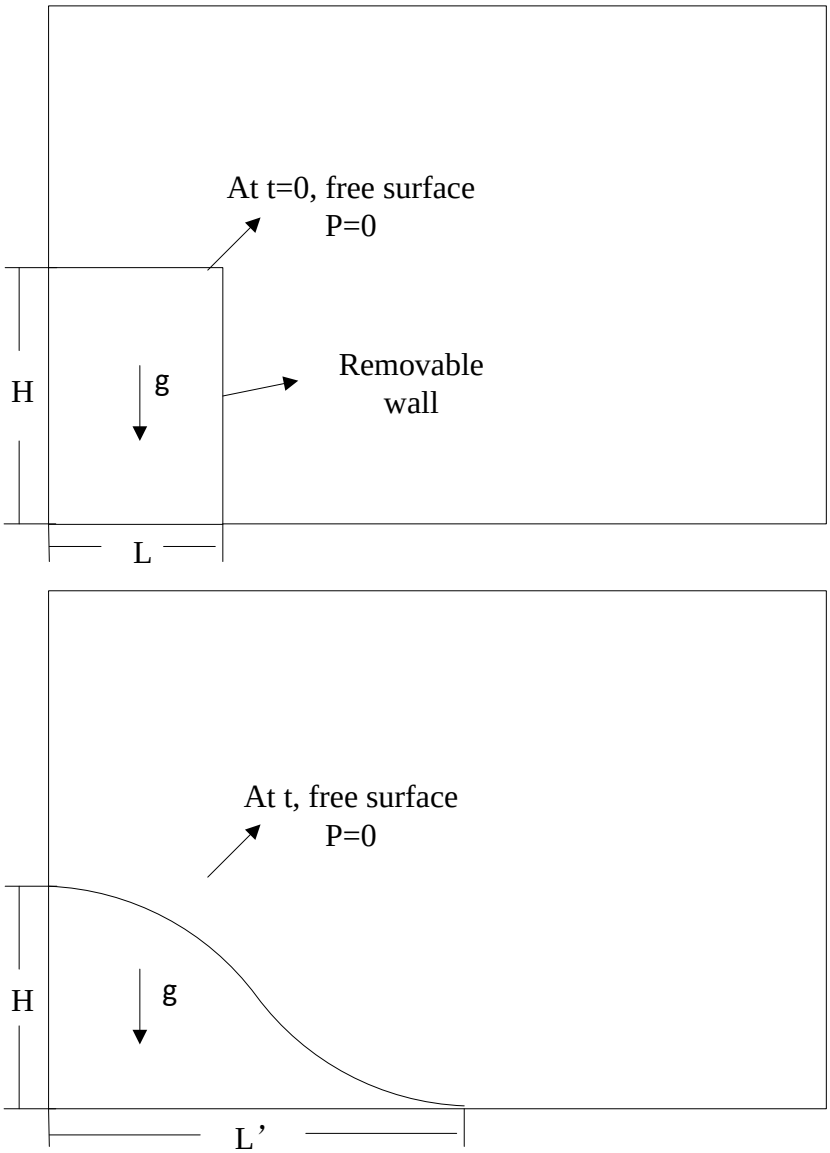


Figure 6.9: Dam breaking model

Simulations are carried out with different kernel functions, time step sizes and number of particles. The particles are assumed to be uniformly distributed in the domain initially. Lennard-Jones form of repulsive force given in Equation (5.1.2) is applied for the left and bottom wall as boundary treatments. Velocity, pressure and position are calculated according to Predictor-corrector algorithm in WCSPH method. Kernels including Cubic spline, quartic, quadratic, Johnson quartic and Gaussian are used. The time step size is $0.5\times10^{-4}s$ and the number of particles is 30 x 60 in these cases. The position of the leading edge of the fluid is recorded and compared with experimental data (Martin & Moyce 1952) as shown in Figure 6.10.

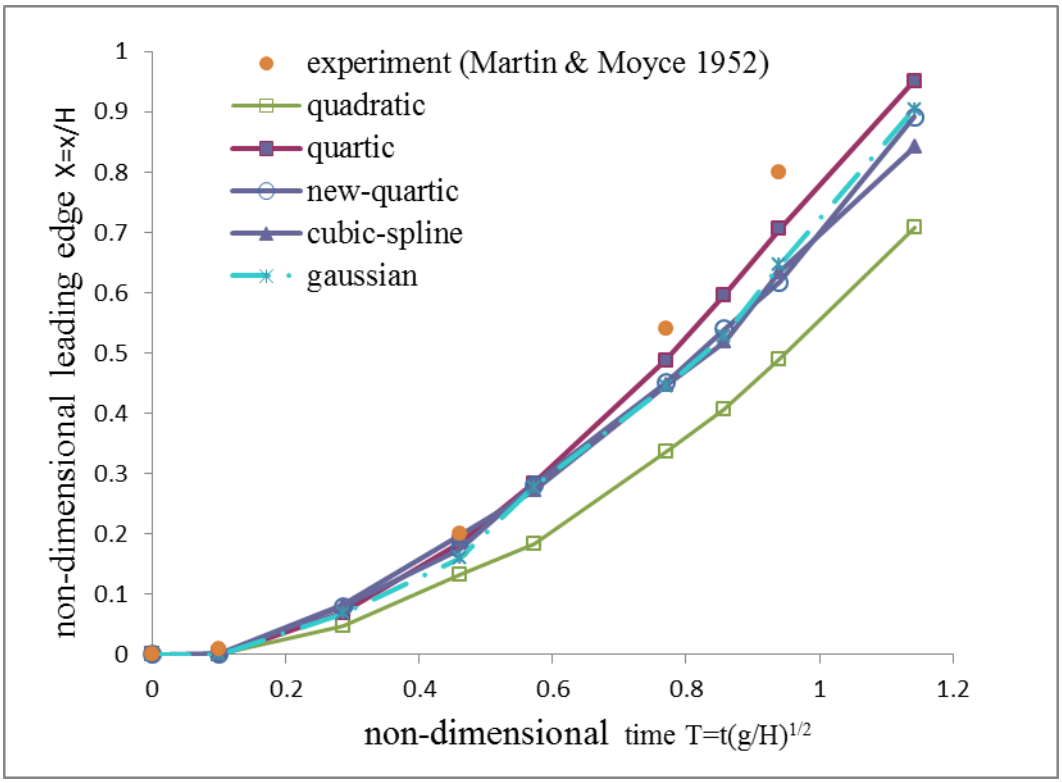


Figure 6.10: Results obtained by using different kernel functions

From Figure 6.10 we can see that the results obtained by using quartic kernel are closest to the experimental data and quadratic kernel produces the worst results. Other kernels produce similar results.

To investigate the stability, various time step sizes are considered. The maximum time step size is decided to be $1.0\times10^{-4}s$ based on the CFL stability condition. New quartic and quartic kernels are utilized in this case with 30 x 60 particles;

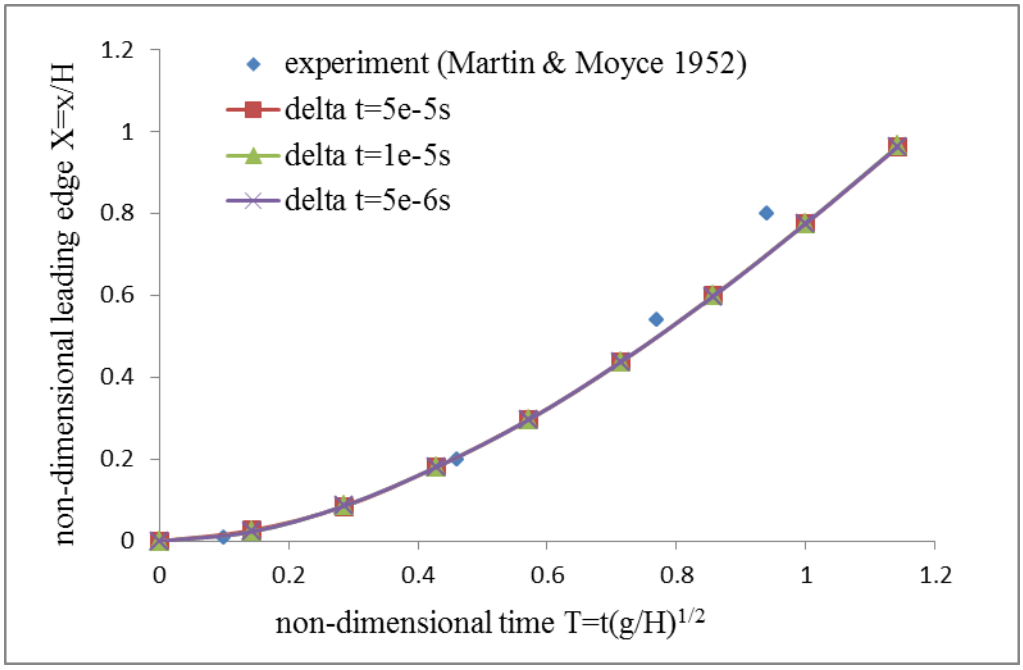


Figure 6.11: Quartic kernel using different time step size

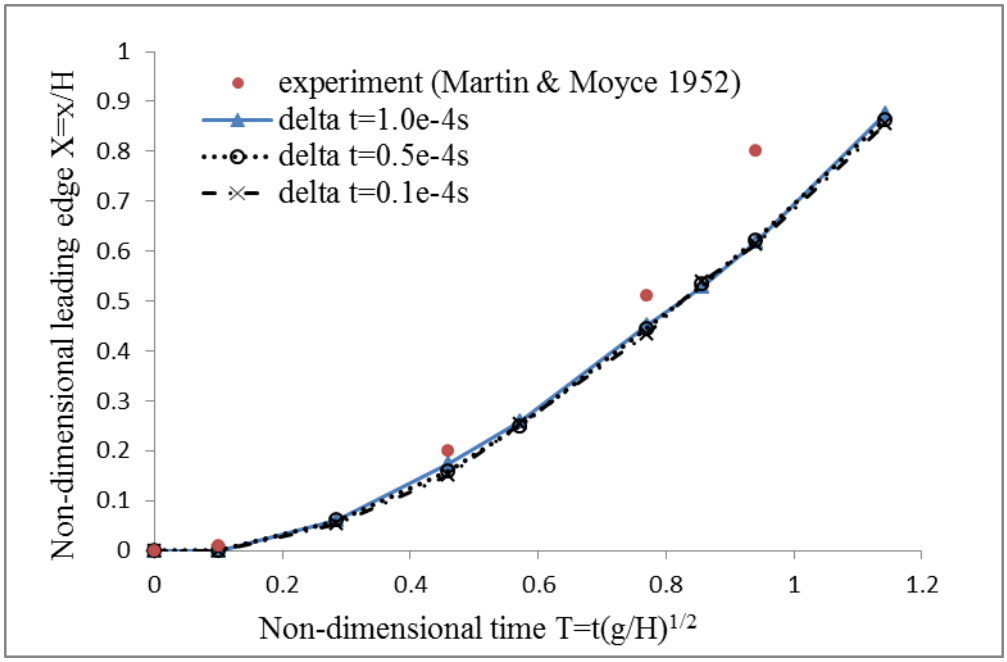


Figure 6.12: New quartic kernel using differetn time step size

It can be seen from Figures 6.11 and 6.12 that time step size does not affect the results significantly. However, during simulation, it was found that when larger time step of $1.0\times10^{-4}\,s$ is used, only the simulations using the new quartic and quartic kernel were successfully completed. For the other kernel functions some inner fluid particles were found to penetrate the boundary with this time step size. This means

that time step size influences the stability of the algorithm and quartic kernel functions have better stability.

Different numbers of particles with quartic and new quartic kernels with a time step size of $0.5\times10^{-4}s$ are used to see the possibility to improve the accuracy by increasing the particle numbers.

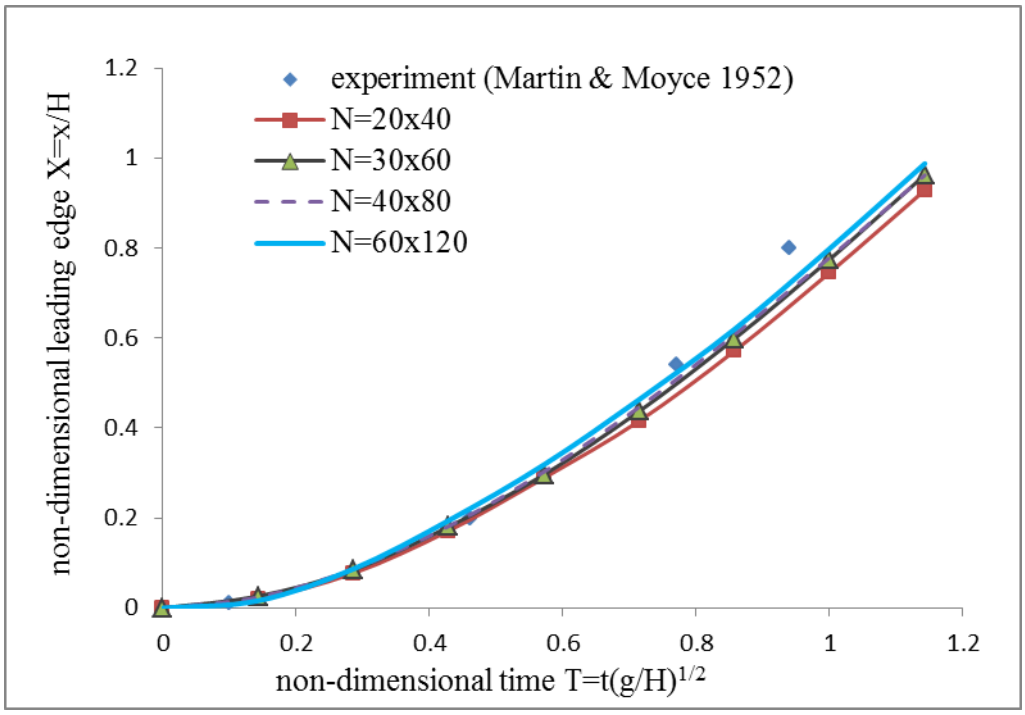


Figure 6.13: Quartic kernel with different number of particles

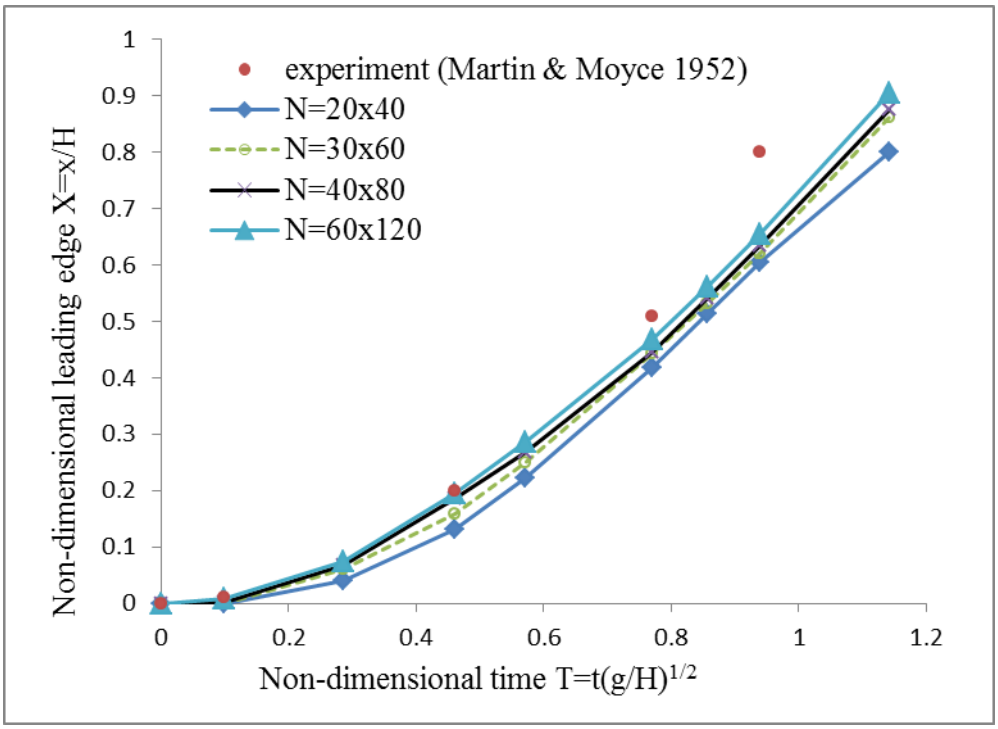


Figure 6.14: New quartic kernel with different number of particles

It is observed from Figure 6.13 and 6.14 that the results are closer to the experimental data with increased number of particles. Especially in the early stage of the fluid motion, a larger number of particles provide better agreement with the experimental data especially in the early stage of the fluid motion. It is expected that a better results can be achieved during the entire process with more number of particles. However, when the particle numbers doubled, the computational time will also increase, and this makes the computation more expensive.

6.3 Summary

SPH approximation has second order of accuracy with smoothing length. Decreasing the smoothing length or increasing the number of neighbouring particles is useful to improve the accuracy. A ratio of 0.5 between particle spacing and the smoothing domain radius, i.e. $\frac{dl}{kh} = 0.5$, can provide accurate results. Different kernel functions may result in some difference for the simulation but generally they produce similar approximations. A dam breaking case study further confirmed that increasing particle numbers can improve the accuracy of simulations. Time step size does not influence the accuracy significantly but it influences the stability of the algorithm.

Although these investigations are based on the WCSPH method, the conclusion is applicable for ISPH method as the general principle is the same. The difference of the performances of these two methods will be discussed in the next chapter.

Chapter 7 Performance comparison of ISPH and WCSPH

As discussed in Chapter 4, the pressure of fluid can be computed through two approaches. One is WCSPH method which uses the equation of state to calculate the pressure and another is ISPH method in which the Poisson's equation is solved. In order to determine which method should be selected for future applications a comparison of the performances of these two methods is carried out using a dam breaking flow simulation as a test case. In ISPH method the fluid density is assumed to be constant. The two solution approaches for Poisson's equation, explicit solution approach and implicit solution approach, are both considered. After the determination of the preferred method, boundary treatments are investigated for the selected method. CPU time and flow patterns obtained from these two methods are compared.

7.1 Dam breaking case 1

The same model for dam breaking case study as presented in Chapter 6 is used again to compare the performance of ISPH and WCSPH methods. Two solution approaches for ISPH methods are both considered. To clarify the difference and to eliminate the influence from other factors such as smoothing length, kernel function choice, particle numbers and so on, same parameters are adopted for these different algorithms. The results of the position of leading edge of the fluid is shown in Figure 7.1

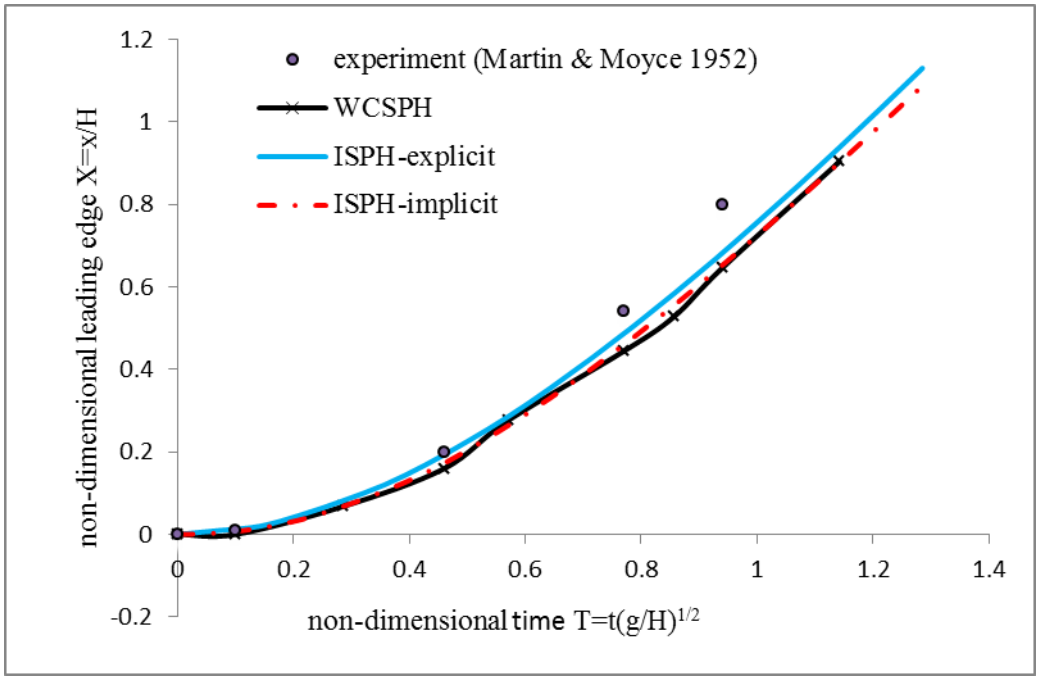


Figure 7.1: Results obtained from WCSPH and ISPH with explicit solution and implicit solution approach

It seems that the results obtained using ISPH method with explicit solution approach are closer to the experiment data although all the numerical results in Figure 7.1 are very similar. In addition to the velocity, to be able to predict pressure correctly is also important. Hence, a simple case with a tank of hydrostatic water is simulated to investigate the pressure predictions of these methods.

7.2 Prediction of hydrostatic pressure

In the previous section the velocity prediction of ISPH and WCSPH is compared. In this section, the pressure distribution produced by these two methods will be investigated using a simple case with a tank of water in hydrostatic state. In order to obtain comparable simulations, the numerical model is set to be the same for these different methods. The tank size is 1.2m wide and 1.0m high, water depth is 0.6m. The particle spacing is 0.01m. Flow patterns at time t=0.5s and t=1s are shown in Figure 7.2 to Figure 7.4, with different colour representing different pressure values.

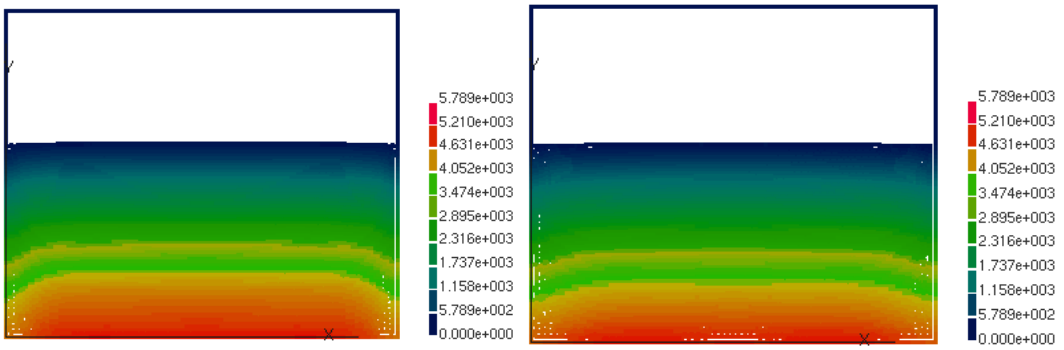


Figure 7.2: Pressure distribution from ISPH with explicit solution approach at 0.5s and 1s

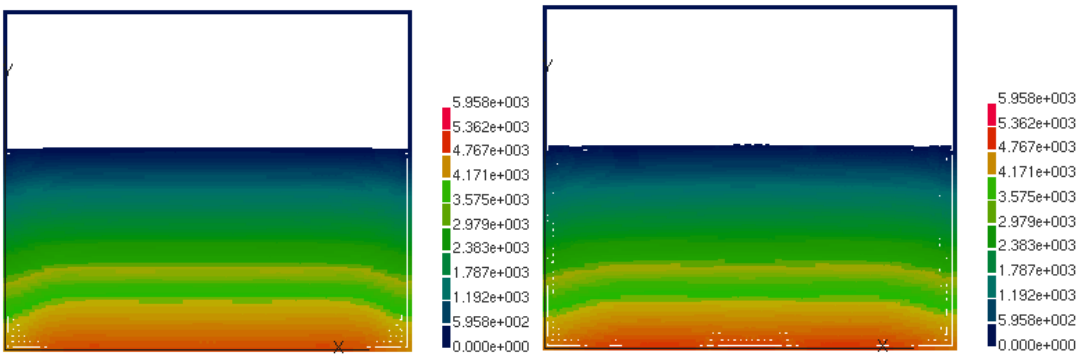


Figure 7.3: Pressure distribution from ISPH with implicit solution approach at 0.5s and 1s

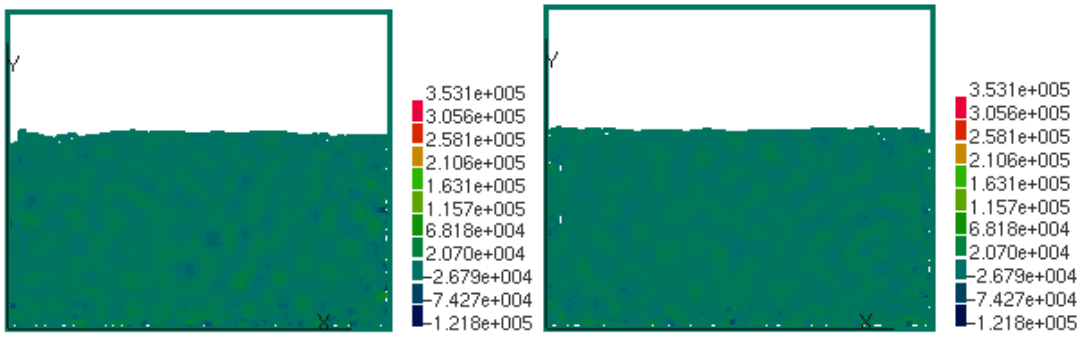


Figure 7.4: Pressure distribution from WCSPH at 05s and 1s

In Figure 7.2, coloured layers representing different pressure values can be observed at different water level. Similarly in Figure 7.3, pressure distribution is clearly defined. By contrast, it is difficult to recognize the coloured layers in Figure 7.4, thus the pressure distribution does not seem to be well predicted. Furthermore, the value range appears to be incorrect as well.

Pressure values of one particle close to the middle point of the bottom of the tank is shown below:

Time (s)	WCSPH	ISPH-explicit	ISPH-implicit
0	5786	5786	5786
0.05	-47020	4215	4500
0.1	2534	5339	4934
0.15	-2416	5005	5042
0.2	43050	4548	5113
0.25	-47890	5060	5167
0.3	-18590	5512	5203
0.35	664.8	5210	5242
0.4	-8680	4811	5271
0.45	13540	5371	5291
0.5	34740	5571	5326
0.55	8442	5036	5351
0.6	-35530	5192	5368
0.65	7605	5630	5382
0.7	-24480	5455	5392
0.75	32560	5036	5410
0.8	-23690	5340	5395
0.85	-21140	5741	5431
0.9	-69940	5316	5435
0.95	7157	5130	5429
1	27060	5544	5406

Table 7.1: pressure values (N/m^2) of a fixed point on the bottom of the tank

In table 7.1, the pressure values obtained using WCSPH method change erraticly at different time and some unphysically negative values are obverved. This is because the pressure is calculated according to the equation of state which strongly depends on the change of density. A slightly reduced value of density may lead to a large negative value of pressure. in contrast, the pressure values obtained using ISPH method with both solution approaches are very stable.

The relative error of pressure is calculated and compared in Figure 7.5

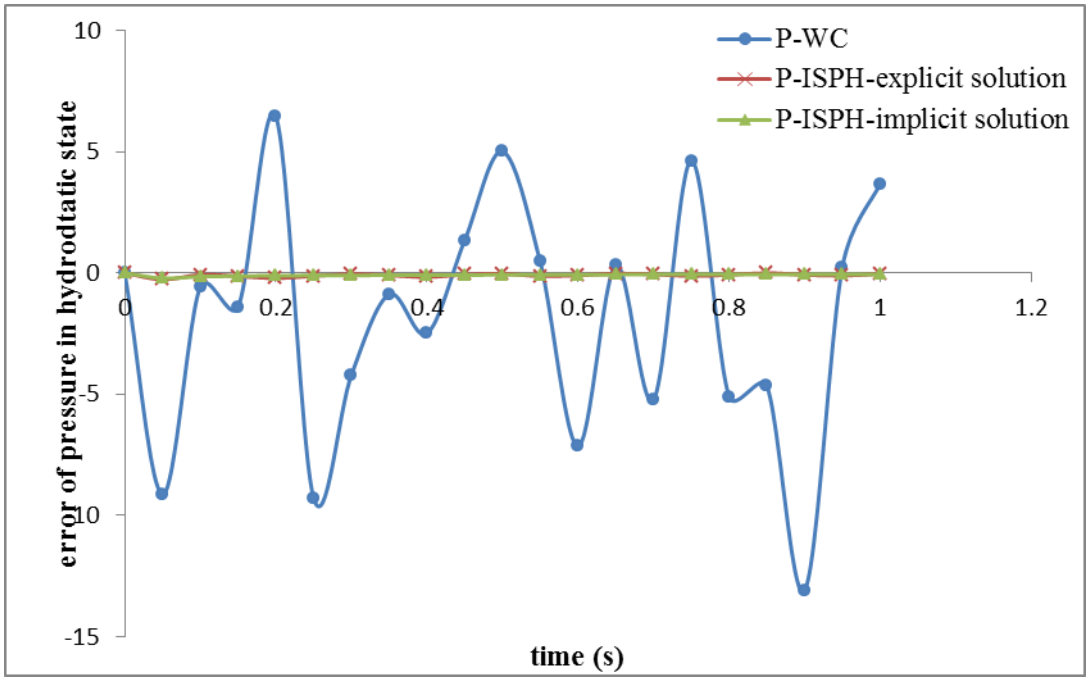


Figure 7.5: Relative error of pressure obtained from different methods in hydrostatic state

From Figure 7.5 it is obvious that the error of pressure values obtained from WCSPH method is high and the values fluctuate irregularly. In contrast, ISPH method with both solution approaches provides smoother pressure values and the error is smaller than 0.01. The CPU time for different methods is shown in table 7.2.

METHOD	Time step size	CPU time per step
WCSPH	0.00001s	0.015584s
ISPH-implicit	0.0005s	0.159695s
ISPH-explicit	0.0005s	0.0281665s

Table 7.2: CPU time for hydrostatic tank simulation of 1s with different methods

From table 7.2 we know that a larger time step size can be allowed for ISPH method regardless of which solution approach is used. The longest CPU time for one time step is using ISPH method with implicit solution approach, which is almost ten times of the time needed in WCSPH method. However, the total CPU time for a simulation of 1s is 319.39s in ISPH and 1558.4s in WCSPH method, which means the total CPU time can be reduced by using ISPH. The shorted CPU time is required when explicit solution approach is applied for ISPH method, which is one fifth of the time required in implicit solution approach.

In order to further confirm that ISPH is able to perform better, another dam breaking case with a barrier in the middle of the tank is simulated in the next section.

7.3 Dam breaking case 2

In this section a dam breaking case with a barrier in the middle of the tank is simulated using WCSPH and ISPH with the implicit solution approach. The model is set up with a tank size of 0.8m wide and 0.6m high, the water column size is 0.3m x 0.3m. Three cases with different number of particles are considered: the first one has 15 x 15 particles; the second one 30 x 30 particles and the third one 60 x 60 particles. The time step size allowed for WCSPH is $1.0\times10^{-5}s$ and for ISPH it is $5\times10^{-4}s$. The motion of water flow at different time is displayed and the pressure distribution is shown with different colours. In addition, CPU time for these two methods is compared.

CASE 1: Using 15 x 15 particles with spacing of 0.02m in WCSPH

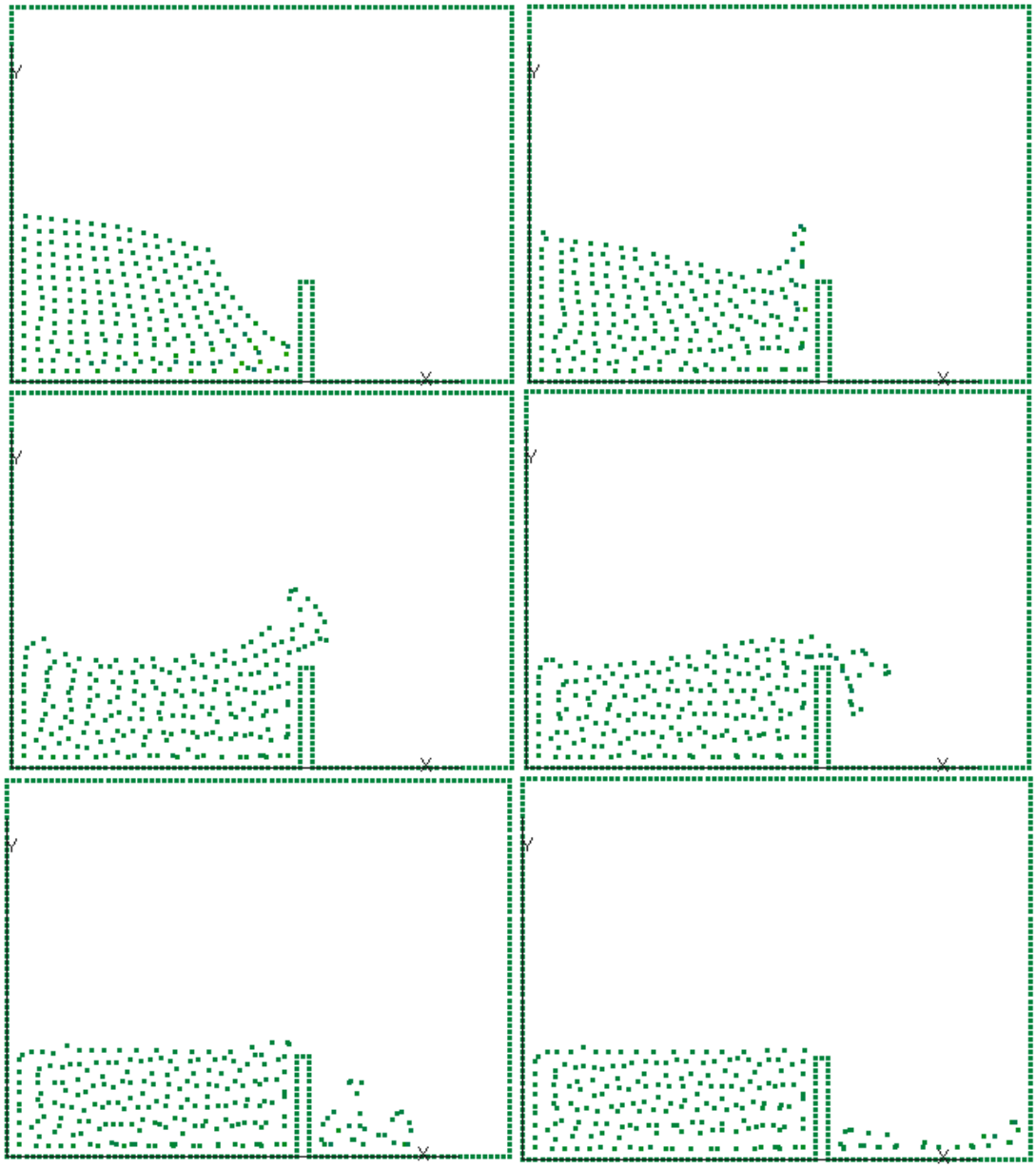


Figure 7.6: Motion of water flow with WCSPH at time 0.15s, 0.25s, 0.4s, 0.55s, 0.8s and 1s using 15 x 15 particles

Using 15 x 15 particles with a spacing of 0.02m in ISPH

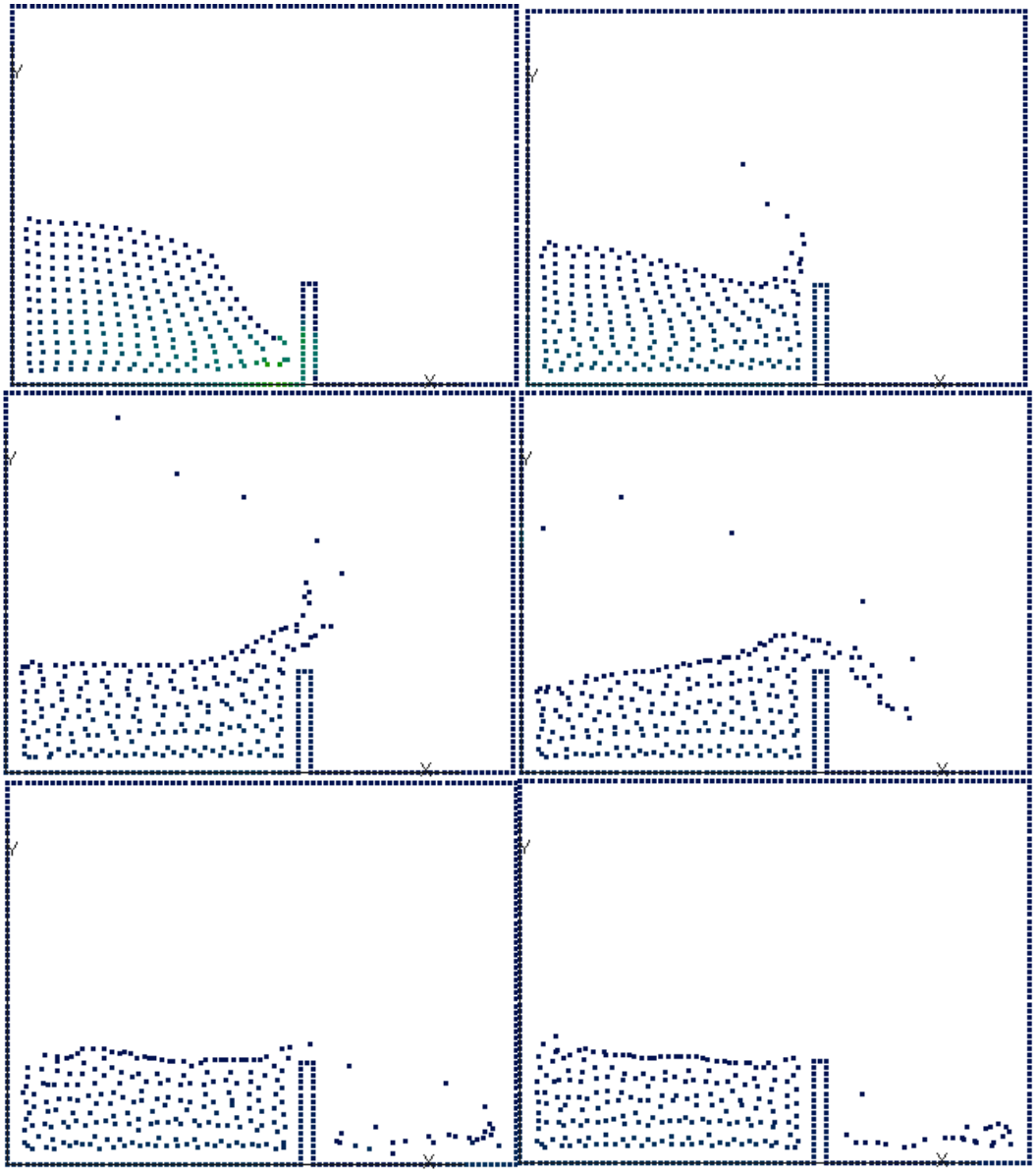


Figure 7.7: Motion of water flow with ISPH at time 0.15s, 0.25s, 0.4s, 0.55s, 0.8s and 1s using 15 x15 particles

Comparing Figures 7.3 and 7.4, the spray generated as the results of water impact on the barrier using WCSPH method is not as violent as that using ISPH. The pressure distribution as indicated by the colour difference of the particles looks more realistic in the ISPH data than in WCSPH results.

CASE 2: Using 30 x 30 particles with a spacing of 0.01m in WCSPH

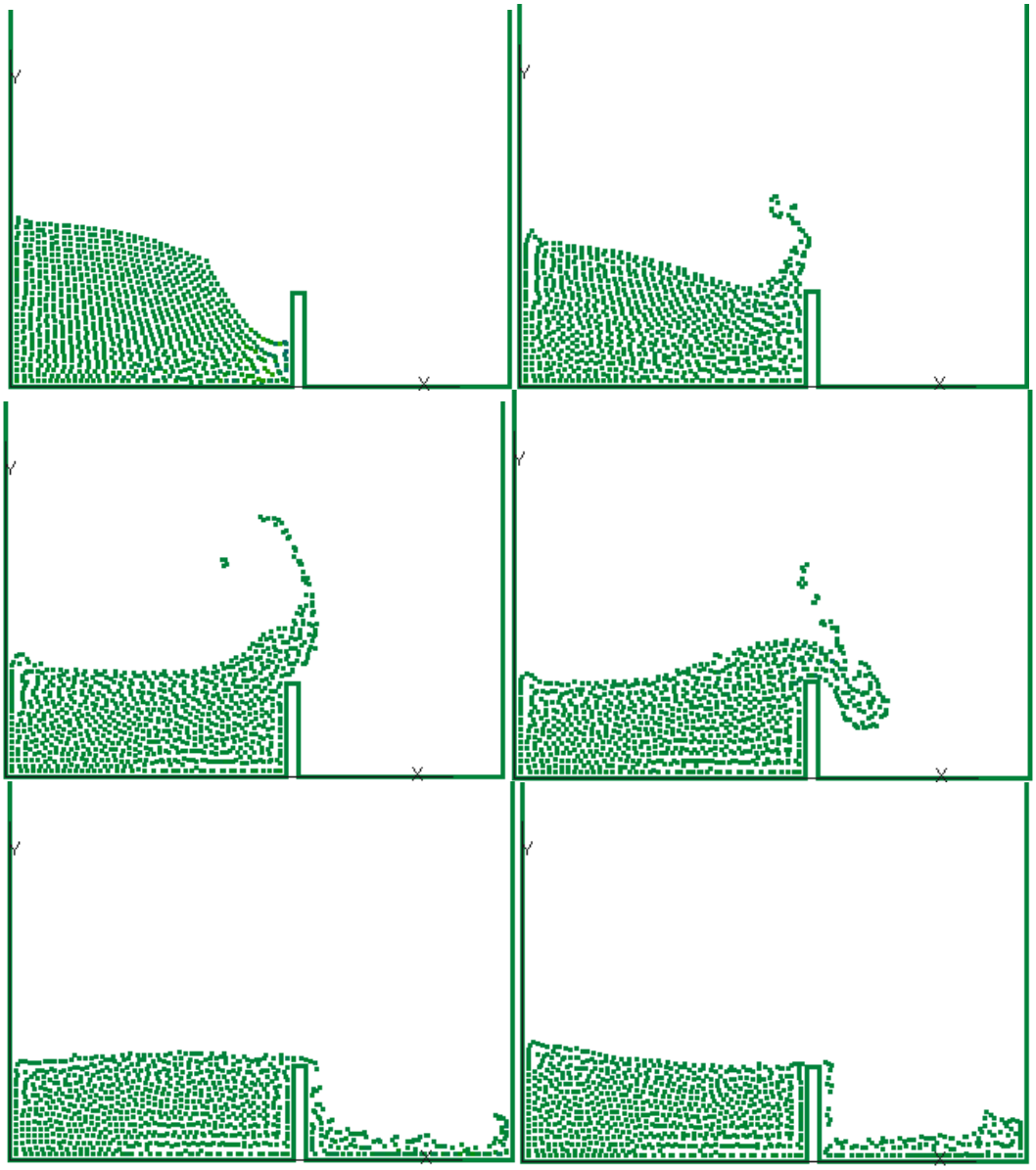


Figure 7.8: Motion of water flow with WCSPH at time 0.15s, 0.25s, 0.4s, 0.55s, 0.8s and 1s using 30 x 30 particles

Using 30 x 30 particles with a spacing of 0.01m in ISPH

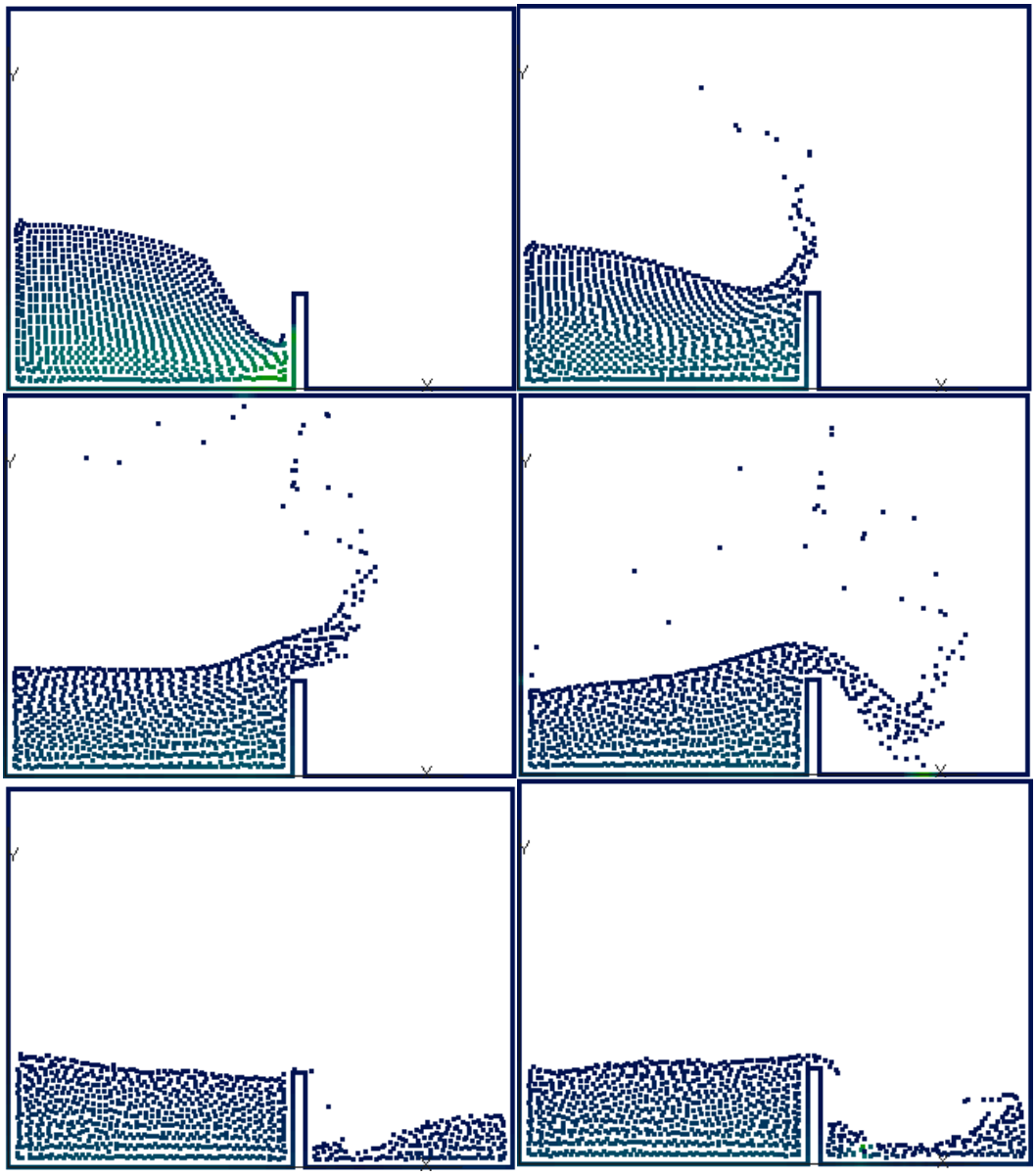


Figure 7.9: Motion of water flow with ISPH at time 0.15s, 0.25s, 0.4s, 0.55s, 0.8s and 1s using 30 x 30 particles

The flow patterns of the fluid in the second case obtained as shown in Figures 7.8 and 7.9 are more realistic than in the first case. This is a further confirmation of the previous conclusion that the results of a simulation can be improved by using more particles. At 0.15s, the flow shown in these two Figures is similar regarding to position and form. At 0.25s, the front of the flow is stopped by the barrier and the rest of the fluid keeps moving because of the gravity. The large difference of the velocity in the fluid causes a splash. As the front of the fluid flowing over the barrier at 0.4s and reaching the other side of the barrier at 0.55s, the difference of the velocity in the rest fluid becomes smaller. At 0.8s, the fluid is divided into two parts. One part is

sloshing slightly behind the barrier. The other part as the front of the flow currently is contained on the other side of barrier. Comparing Figure 7.8 and Figure 7.9, some obvious differences of the flow patterns can be seen. The pattern of the splash generated in ISPH method is more violent than in WCSPH method. In ISPH method some of the particles fly like a fragment of the spray while in WCSPH simulation it seems that the particles of the spray are held together. The cause of this is likely the pressure difference on the free surface of these two methods. In ISPH method, pressure is enforced to be zero for the particles on the free surface as a boundary condition whereas in WCSPH method, pressure values are calculated using the equation of state. Hence the pressure is not necessarily zero on the free surface in WCSPH. At 1s, there is more fluid over the barrier in ISPH than in WCSPH simulation. This indicates that the velocity of the fluid obtained in ISPH method is larger than WCSPH.

Looking at the colours of the particles which represents pressure distribution, it seems that only one single colour is shown in WCSPH which means the pressure is almost the same everywhere while in ISPH it can be seen that the colour on the bottom is brighter which means a higher pressure is in the deeper water. Hence, pressure distribution produced by using ISPH method is more reasonable.

CASE 3: Using 60 x 60 particles with a spacing of 0.005m in WCSPH

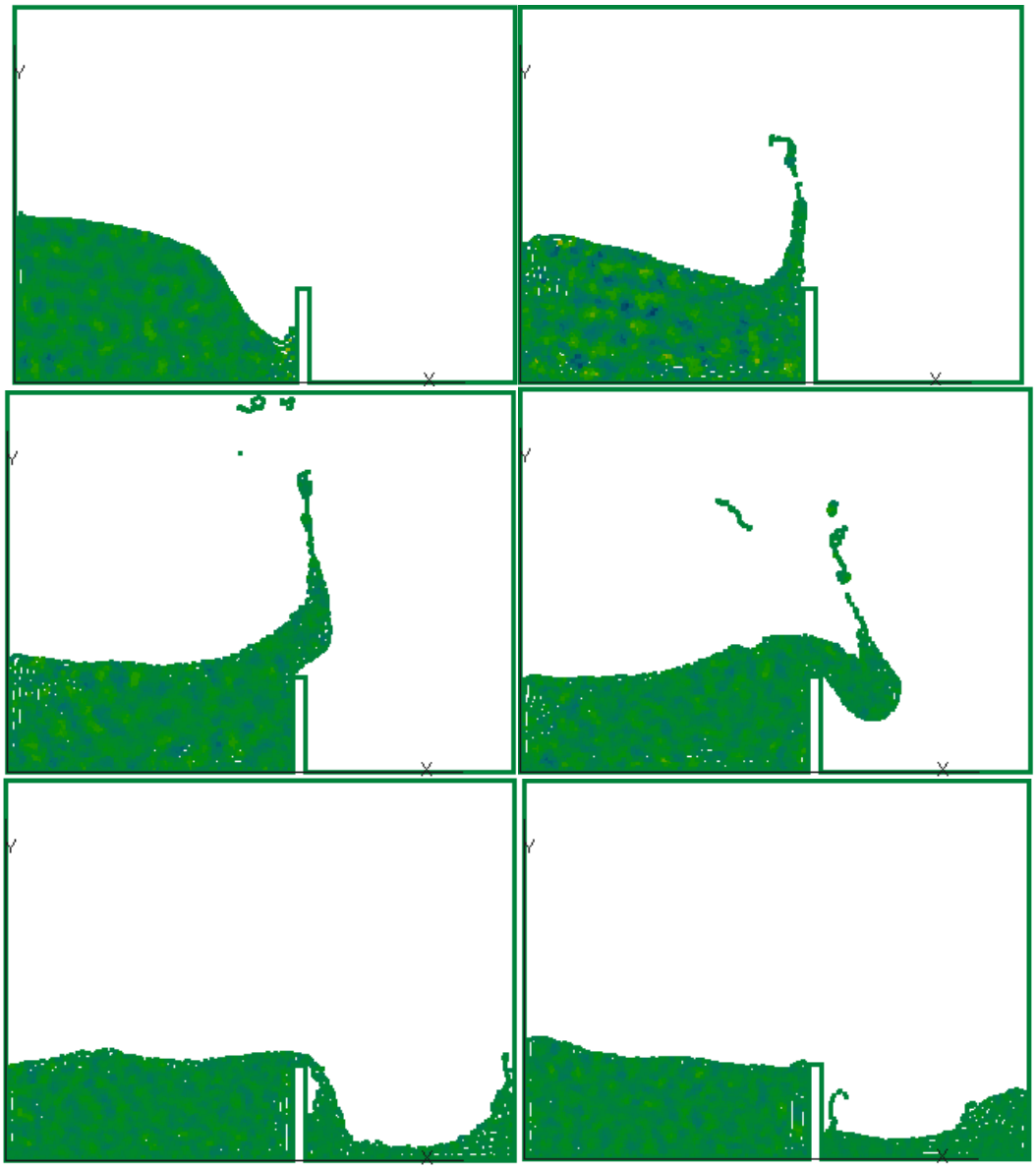


Figure 7.10: Motion of water flow with WCSPH at time 0.15s, 0.25s, 0.4s, 0.55s, 0.8s and 1s using 60 x 60 particles

Using 60 x 60 particles with a spacing of 0.005m in ISPH:

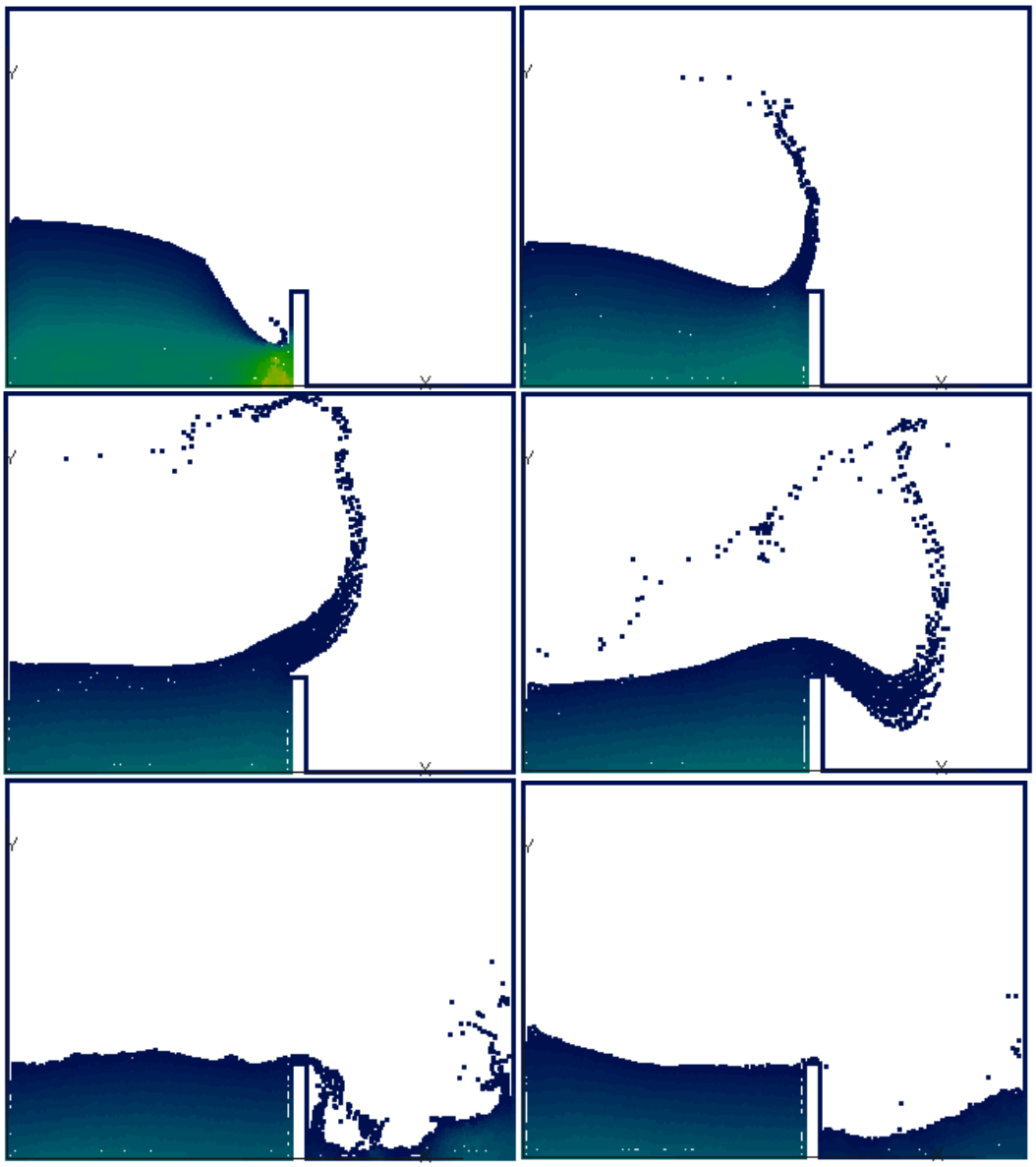


Figure 7.11: Motion of water flow with ISPH at time 0.15s, 0.25s, 0.4s, 0.55s, 0.8s and 1s using 60 x 60 particles

The flow patterns of the flow and spray shown in Figure 7.10 and 7.11 are much clearer than the previous two cases. Same as the second case, the results produced by using WCSPH method are still different from the results produced by using ISPH method. Several levels of different colours can be seen in Figure 7.8, which represents different pressure values at different water level. In contrast there is no clear colour level shown in Figure 7.7. This means that the pressure values obtained by using ISPH are more reasonable and better than WCSPH.

The CPU time for these three cases is given in table 7.3

spacing (m) / SPH method	dl=0.02m	dl=0.01m	dl=0.005m
WCSPH	199.16(s)	246.17(s)	933.29(s)
ISPH	7.471(s)	28.326(s)	124.6(s)

Table 7.3: total CPU time used with WCSPH and ISPH method by using different particle spacing (Time step size used is $5\times10^{-4}\,s$ for ISPH and $10^{-5}\,s$ for WCSPH)

It can be concluded from the figures above from 7.5 to 7.11 that a more realistic motion can be obtained by using more particles for both WCSPH and ISPH methods. To achieve a comparable simulation fewer particles are needed in ISPH method than in WCSPH method. Pressure distribution obtained from ISPH method is more accurate than WCSPH method. Considering the CPU time required for these two methods, ISPH is faster.

In conclusion, with the same algorithm process ISPH method can produce more accurate pressure values than WCSPH and also it requires less CPU time. Therefore, in this project ISPH method is preferred. The explicit solution approach for ISPH method is able to provide similar accuracy with the conventionally used implicit solution approach but the CPU time required is much less than the implicit solution approach.

The boundary treatment is now investigated for ISPH method to ensure easy model generation and efficient simulation.

7.4 Boundary treatment investigation

Two boundary treatments were proposed for ISPH method in Chapter 5. Although they have been used in WCSPH before, it is the first time they are used in ISPH method. These two boundary treatments are easy to implement and they are expected to be efficient for fluid structure interaction problems. They are applied to a dam breaking case to investigate the pressure calculations.

In this case, the overall height of water column is set to be 0.6m and the width is 1.2m. The size of the solid container is 3.22m long. The initial spacing of fluid particles is 0.01m, smoothing length is determined as 1.33 times of the spacing. The pressure values at a point 0.16m from the bottom on the right wall are recorded. The results are obtained using ghost particle treatment, half spacing of wall boundary particles treatment and repulsive force boundary treatment and they are compared with experimental data (Zhou, et al. 1999) and the numerical data produced using Navier-stokes solver (Abdolmaleki, et al. 2004) .

Analysis of Pressure values

Results obtained from different boundary treatments with SPH methods are compared with experimental data as shown below in Figure 7.12.

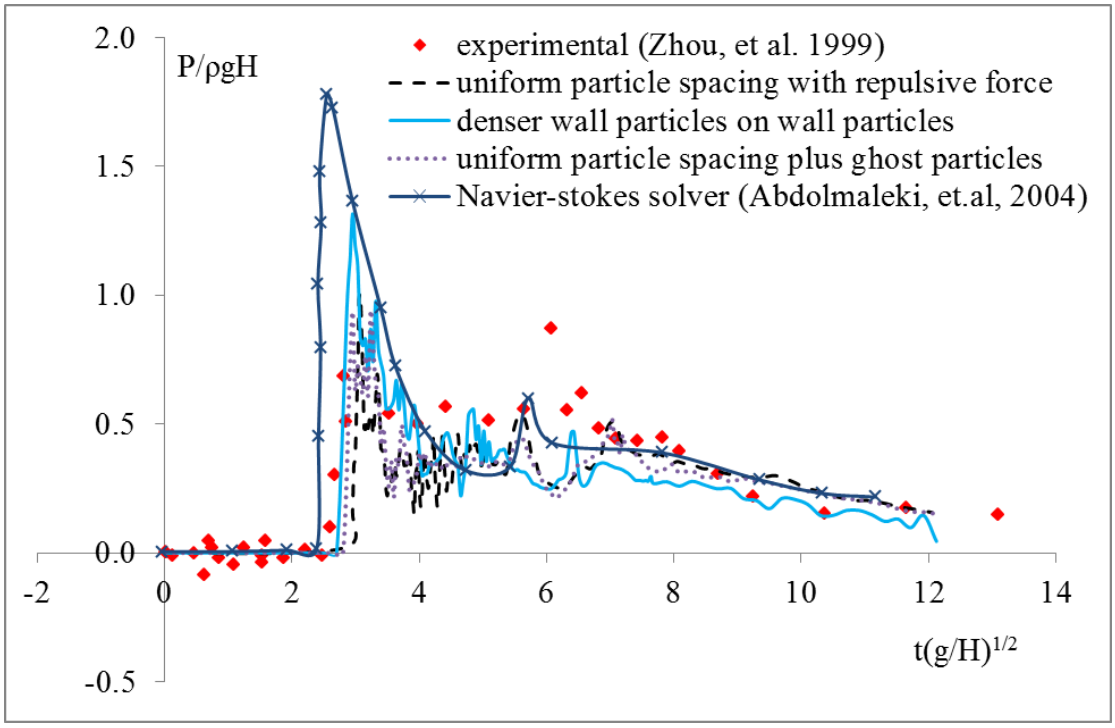


Figure 7.12: Pressure history at point (3.22, 0.16) from different boundary treatments

From Figure 7.12 it is clear that all these three boundary treatments provide the first pressure peak around the right time compared with experimental data. The first peak values obtained from a Navier-Stokes solver are slightly higher than the observed in the experiment. Boundary treatments with repulsive force and ghost particles produce similar results and a denser wall particle boundary treatment gives slightly higher

peak value than the other two treatments. The overall curves agree well with experimental data except the second peak value. There is no obvious second peak pressure in the simulations. But compared with other numerical methods such as Navier-Stokes solver, SPH gives values closer to the experimental data.

Investigation of using different numbers of particles with these two boundary treatment is also carried out and the results are shown below in Figure 7.13.

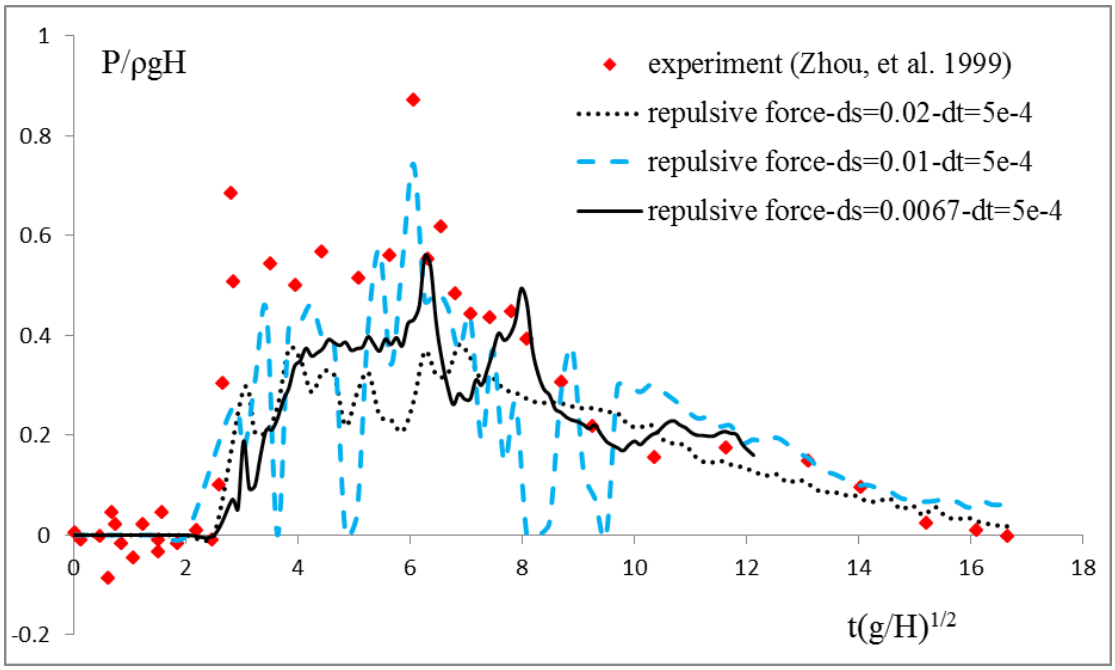


Figure 7.13: Investigation of repulsive force treatment with different particle spacing

From Figure 7.13 it can be seen that the curve with particle spacing of 0.0067m is closer to the experimental data. The values obtained with particles spacing of 0.01m fluctuates more than the others, which implies that decreasing the particle spacing does not always improve the accuracy. Since the ratio between particle spacing and smoothing length is fixed, so the number of neighbouring particles of a particle is fixed but the smoothing length is reduced accordingly. Decreasing smoothing length should lead to an improvement of the accuracy according to the previous investigation. This may imply that with the particle spacing reduced the number of particles will increase and the time step size should decrease to ensure results convergence.

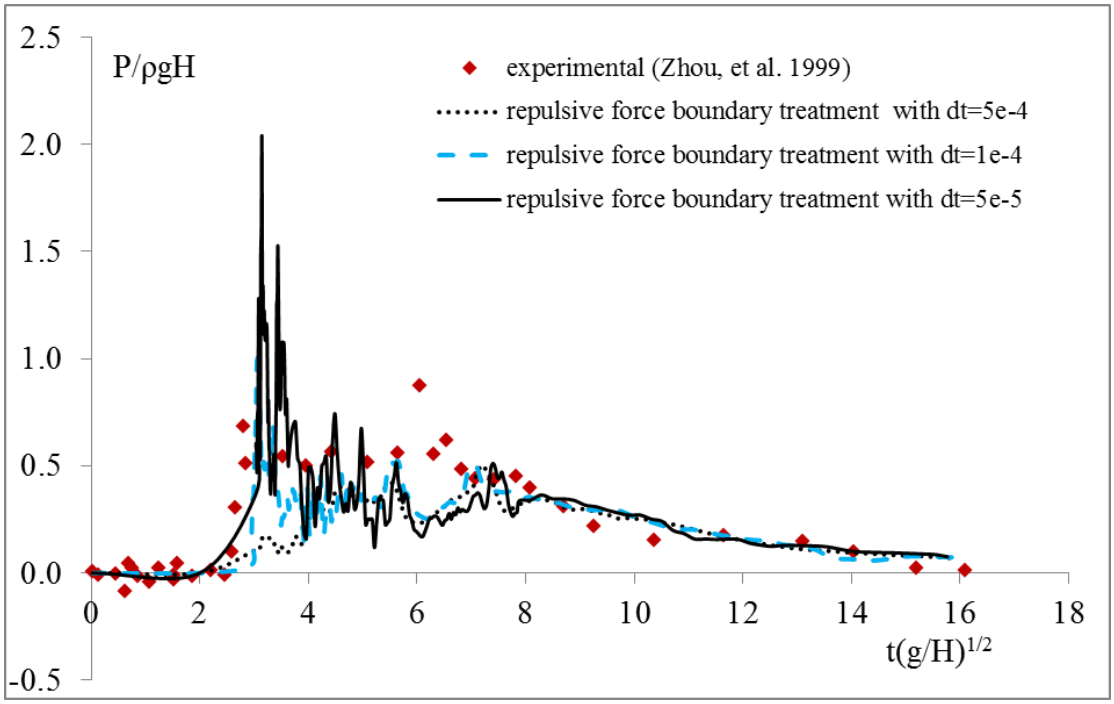


Figure 7.14: Investigation of repulsive force boundary treatment with different time stepping sizes

Figure 7.14 shows the difference between the curve representing the results using time step size of $5\times10^{-4}s$ and the other two curves especially in the predicted period of the first peak. The rest of the curves are almost the same and this means all the time step sizes are appropriate. When time stepping size of 0.0001s is used, the results are close enough to the experiment data. Considering the results shown in Figure 7.13, a time stepping size of 0.0001s seems to be a better choice for the simulation. However, with smaller time step it seems that the values of pressure fluctuate more severely than using a larger time step. This may imply that there are some other factors which influence the convergence such as the disturbance from the boundary treatment.

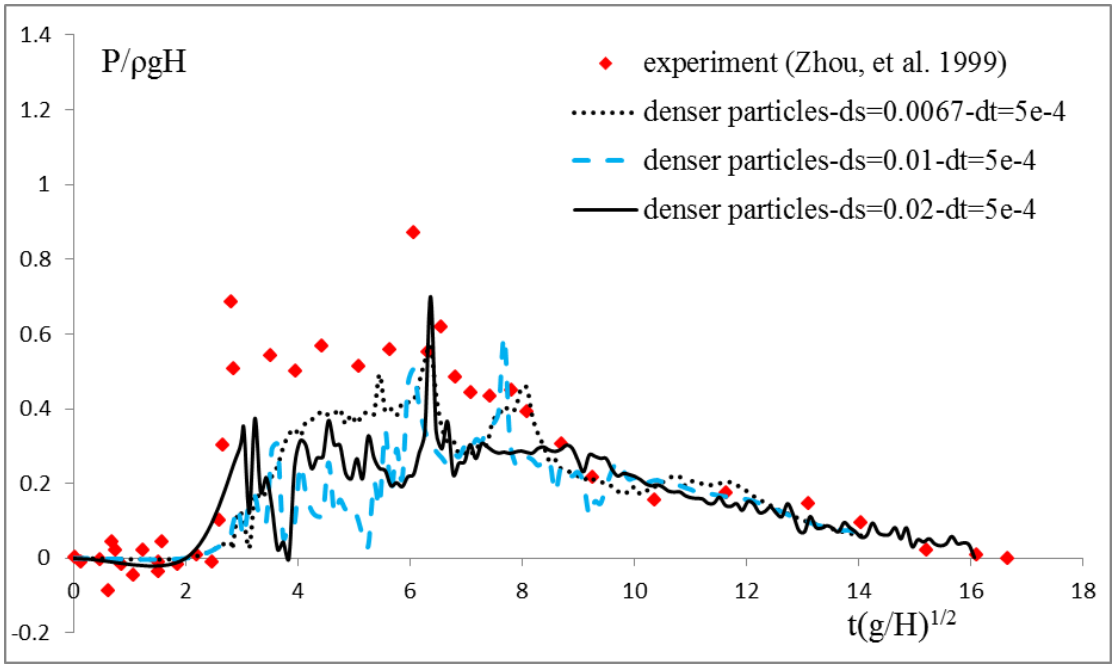


Figure 7.15: Investigation of denser wall particles treatment with different particle spacing

Comparing Figure 7.15 with Figure 7.13, similar observation can be made. The results obtained using a particle spacing of 0.01m seem to be worse than the one obtained using a spacing of 0.02m. To clarify this, the effect of using different time step sizes is studied and the results are shown below in Figure 7.16.

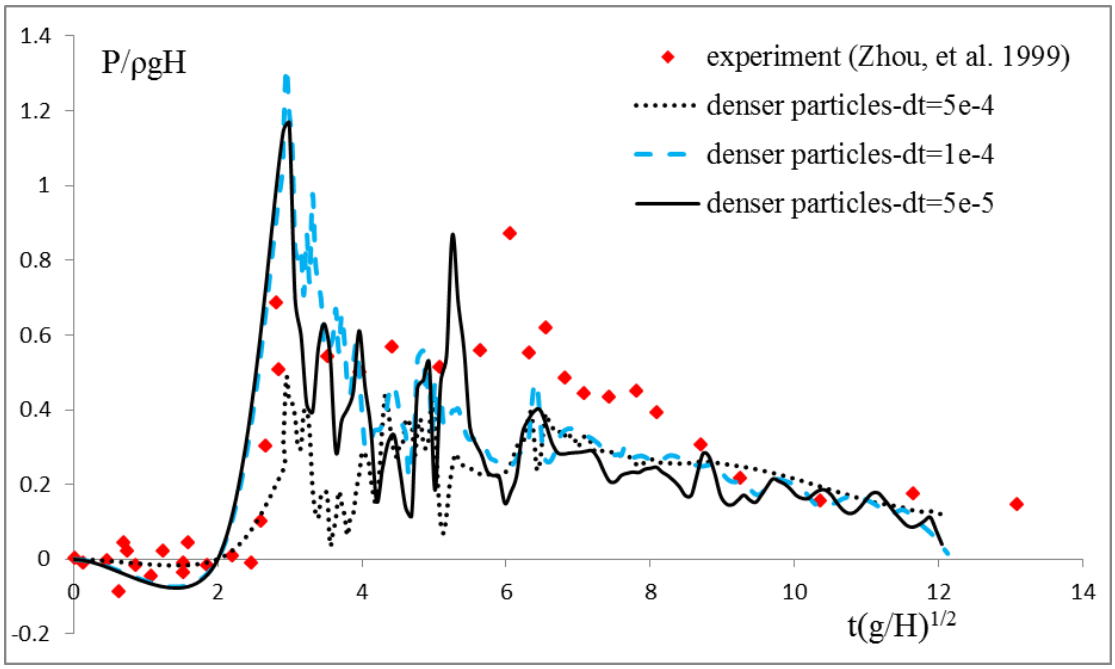


Figure 7.16: Investigation of denser wall particles treatment with different time stepping sizes

From Figure 7.16 we can see that similar to the repulsive force treatment, using a time step size 0.0001s provides better results than using 0.0005s. Only small difference can be observed when the time step size is further reduced. In this case a second peak can be observed when time step size of $5\times10^{-5}s$, which may indicate that using denser wall particles is a better boundary treatment compared to repulsive force treatment.

7.5 Summary

From the results of dam breaking simulations it is clear that ISPH produces better results than in WCSPH method. Another dam break case with a barrier in the middle of the tank was studied to further investigate the difference of the performance of these two methods. Water flow motions and the pressure at different times with different particle numbers were compared. Pressure distribution and water motion can be captured more realistically in an ISPH simulation. Similar performances are observed by using the two solution approaches for ISPH method. The explicit solution approach has a great potential for ISPH although it is not conventionally used. The CPU time required needed in using WCSPH. Hence, ISPH is preferred. As the traditional boundary treatment for ISPH requires ghost particles, it is hard to use for complex geometries, two boundary treatments including repulsive force and denser wall particles are investigated. Pressure values obtained using these two boundary treatments are studied. The results show that these two boundary treatments work well with ISPH method. Applications of ISPH method to fluid solid interaction and multi-phase flow will be discussed in the following chapters.

Chapter 8 Application to 2D fluid-rigid body interactions

Based on the outcome of the investigation of different SPH methods in the previous chapter, ISPH is selected for the simulation of incompressible fluid. And two boundary treatments have been found to be suitable to produce good pressure values with ISPH method. They should be efficient for the modelling of the interface of fluid structure interaction problems. In this chapter, two fluid rigid body interaction examples, water entry of a wedge and the dam breaking problem involving a spring supported rigid wall, are simulated to demonstrate the performance of this method. At the interface, the solid is treated as a wall boundary to ensure no penetration. The physical properties carried by the solid particles contribute to the corresponding quantity calculations of the inner fluid particles within the smoothing domain. Therefore, the influence of the solid particles on the inner fluid particles is considered in the particle approximation, and the coupling condition between the fluid and the solid is automatically satisfied. The force acting on the solid is determined by the summation of the pressure on all the solid particles.

8.1 Wedge dropping simulation

Water impact is a common problem found in marine and offshore engineering. Wedge dropping test is used to study the reaction of ship slamming. The understanding of the influence of the fluid on the body is of interest for the safety consideration in the design of the marine structures.

As the velocity of the dropping wedge depends on the interactions with fluid, simulation could be difficult for the grid based methods due to the treatment of free surfaces and moving solid boundaries. Oger, et al. (2006) applied WCSPH method to wedge dropping simulation using denser particles in the impact area and the smoothing length was changed depending on the requirement of accuracy to ensure an acceptable level of density fluctuation in the fluid. Gong, et al. (2009) proposed an alternative method by using a sponge layer on the bottom of the tank to adjust the density calculation of the fluid particles. When Shao's ISPH method was applied to water entry of free-falling wedge, mirror particles were used on the moving solid

(Shao 2009). With Lee's ISPH method, the proposed two boundary treatments can be applied which will simplify the model generation and reduce computation time.

In this section, a symmetric wedge with a dead-rise angle θ of 30° (an angle measured upward from a horizontal plane at keel level) dropping into water as shown in Figure 8.1 is simulated using ISPH with the denser wall particle boundary treatment. The weight of the dropping wedge is 241kg, the width of the wedge is 0.5m and the length is 1m. The tank size is 2m x 1m. In the simulation, the wedge is placed just above the free surface of the calm water with a dropping velocity of 6.15m/s given from the 2D experiment (Zhao, et al. 1997).

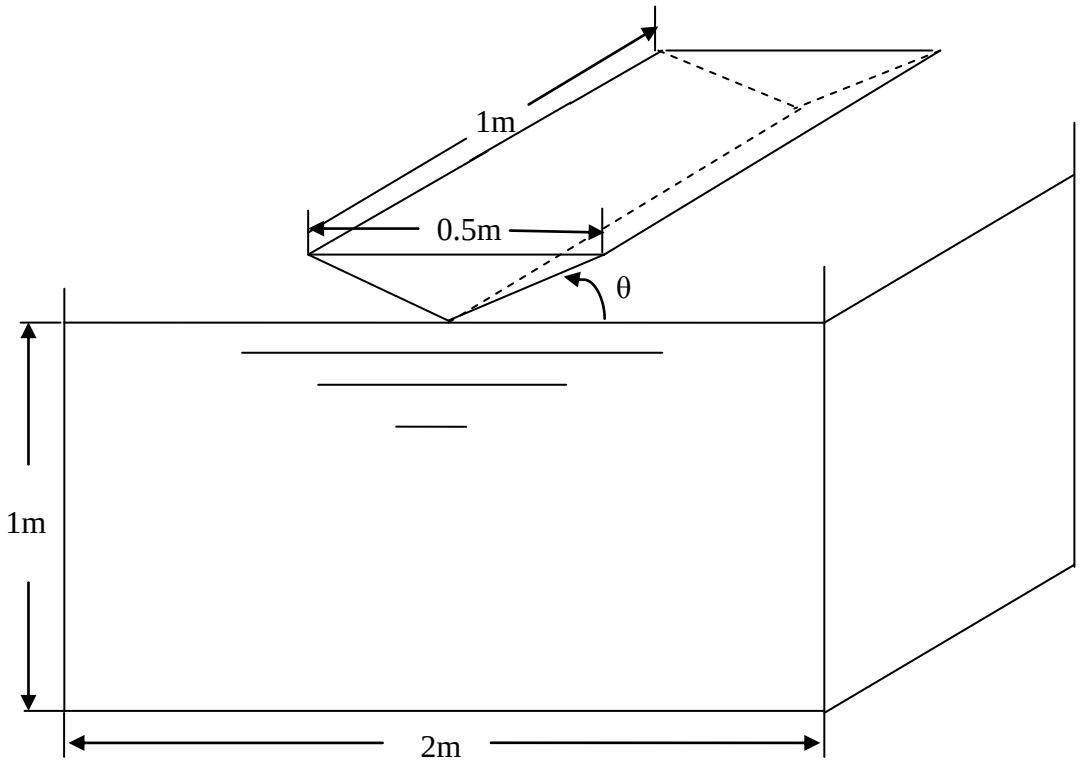


Figure 8.1: Water entry of wedge

The wedge is only allowed to move in the vertical direction and its motion follows the equation of motion for rigid body. The resultant water pattern is compared with the photo from the experiment (Tveitnes, et al. 2008).

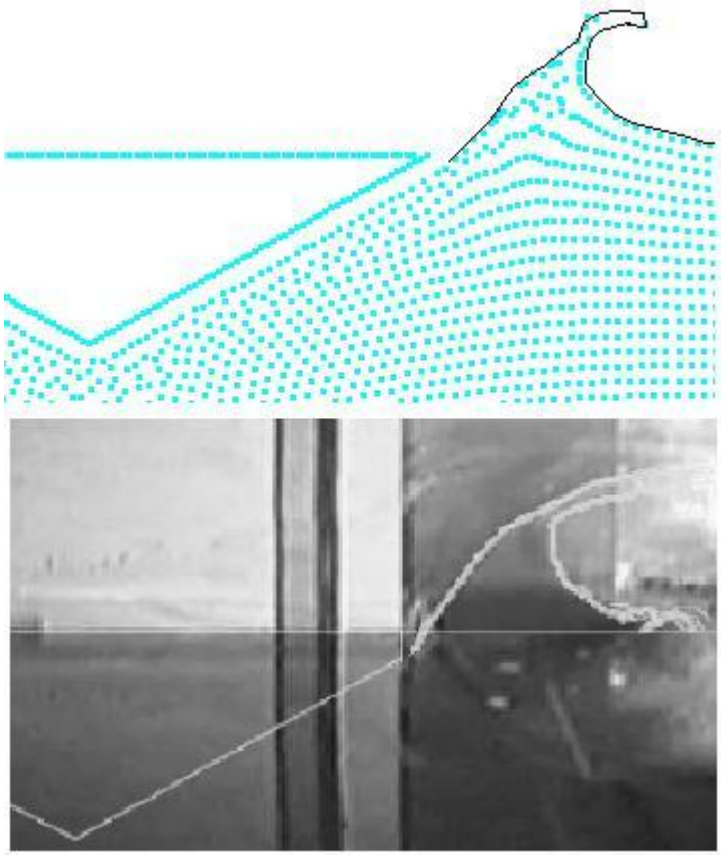


Figure 8.2: Water pattern during wedge dropping

The vertical acceleration of the wedge will decrease when it entries the water. When the falling wedge hits on the water surface, the surface will be break because of the strong impact and water is pushed up around the wedge and water jets are generated in this stage as shown in Figure 8.2. The wave pattern obtained from SPH simulation is remarked with black lines for easy compare with the experiment. The velocity of the dropping wedge after impact and the impact forces on the wedge from water are compared with experimental data and analytical result given by Zhao, et al. (1997) in Figure 8.3 and 8.4 respectively.

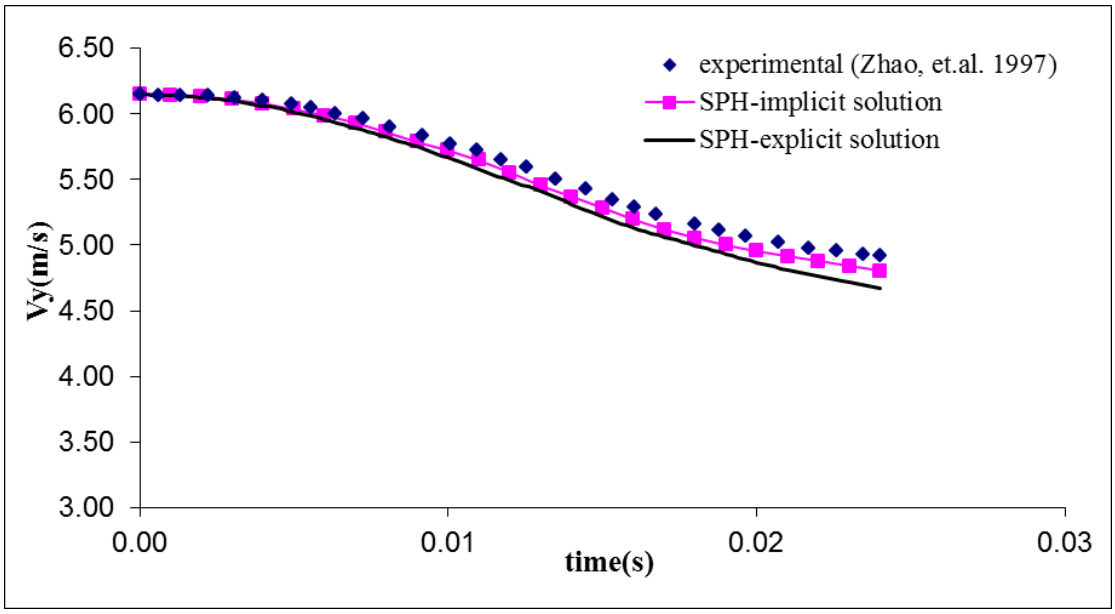


Figure 8.3: Wedge dropping velocity in the water

Figure 8.3 shows a good agreement between velocity values obtained from experiment and the current SPH method.

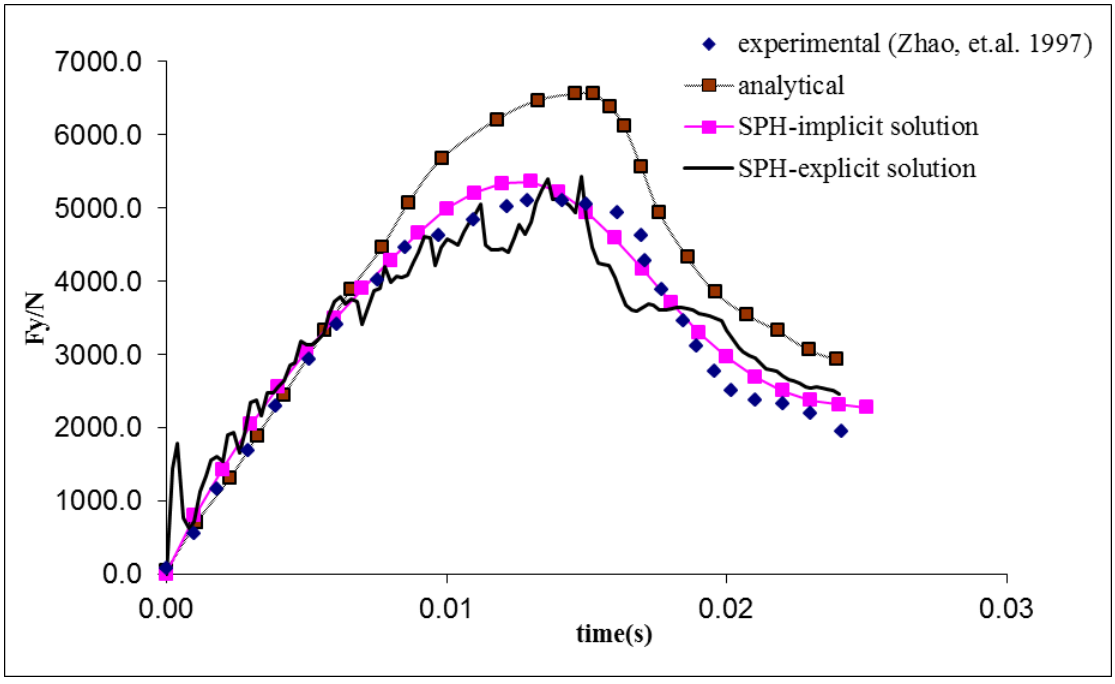


Figure 8.4: Impacting force on wedge

In Figure 8.3 it shows that the dropping velocity decreases more rapidly after 0.007s. Initially, the fluid force increases steadily and then slows down before reaching the peak at around 0.015s. After that the force starts to decrease. The dropping velocities obtained based on SPH method are slightly lower than the experimental values in the later stage with a maximum error of 2%. For the vertical fluid force shown in Figure 8.4, the computed values are slightly over-predicated at first and then it is under-predicated for a short period of time, but it is higher than experimental values at the last stage. Overall, both the dropping velocity and vertical fluid force obtained from the proposed SPH method agree well with experimental values. To investigate the effect of different parameters on the water entry process the following 3 cases are studied

Case 1: different wedge masses with dead-rise angle of 30° and initial dropping velocity of 6.15m/s

Case 2: different initial dropping velocities with wedge mass of 241kg and dead-rise angle of 30°

Case 3: different dead-rise angles with wedge mass of 241kg and initial dropping velocity of 6.15m/s

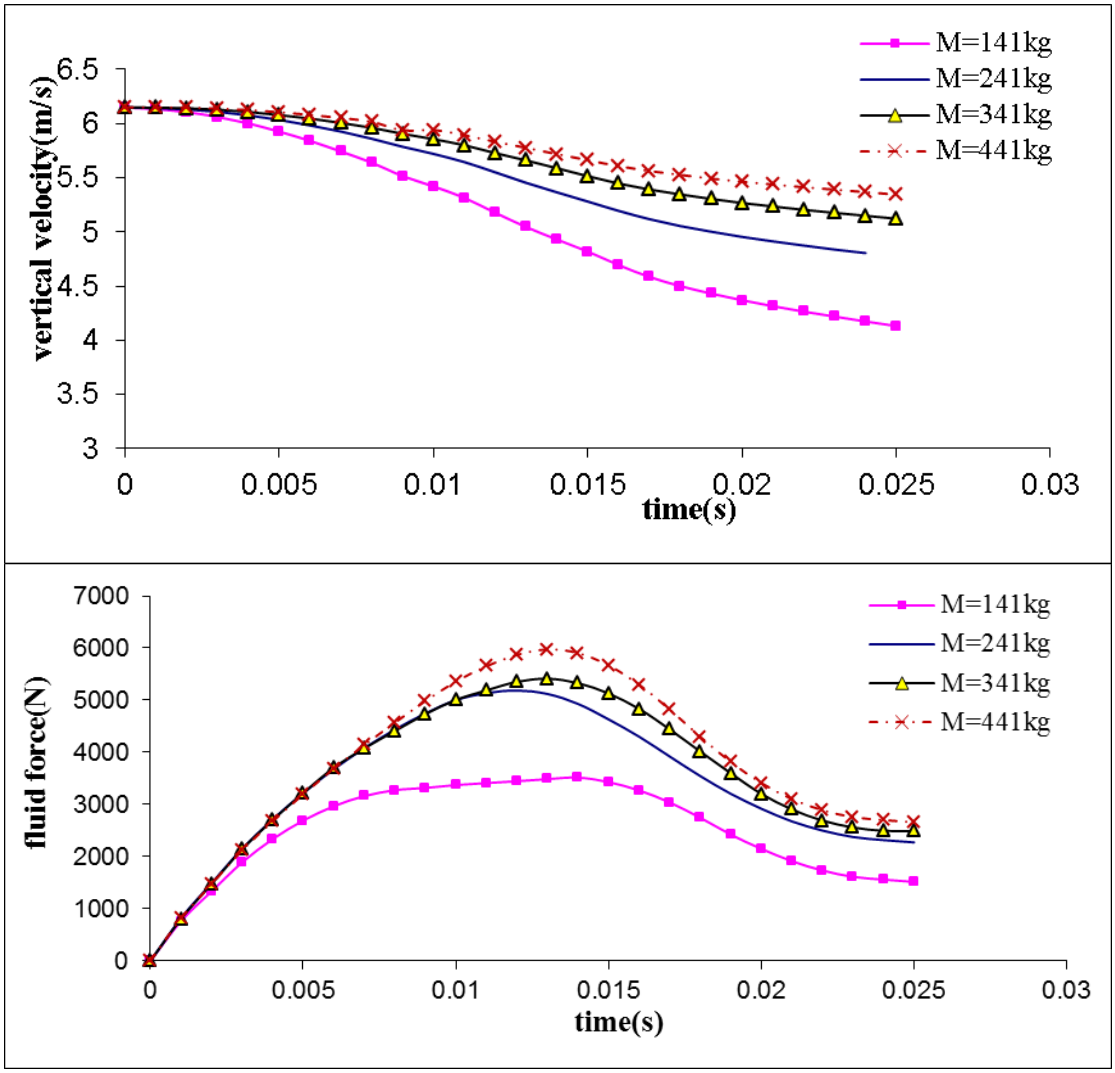


Figure 8.5: Time history of dropping velocities (top) and fluid forces (bottom) for different wedge mass

From Figure 8.5, it is clear that the heavier the wedge is, the slower the velocity decreases with higher peak fluid force. This is because the initial velocity is in the same direction as gravity force and the gravity force is larger for a heavier weight. So the results are reasonable.

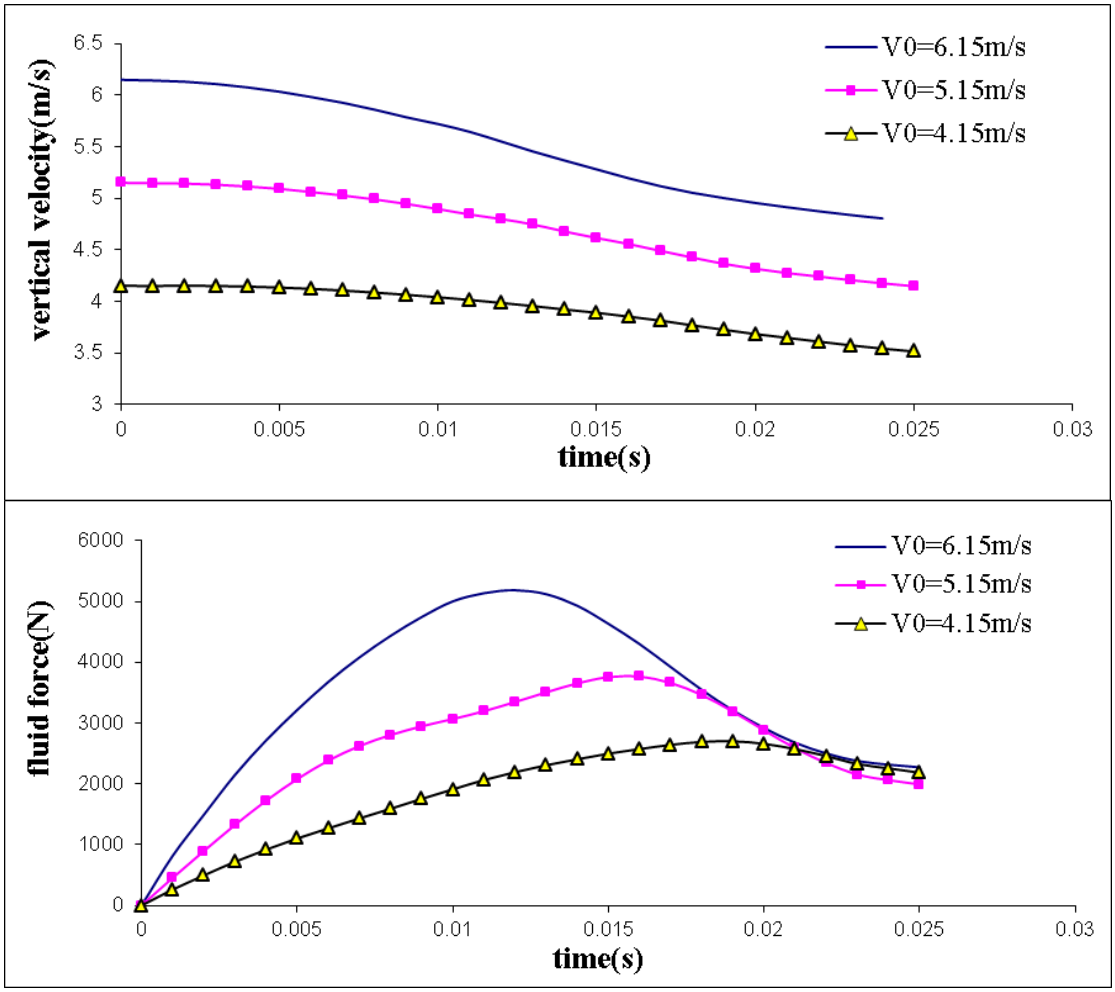


Figure 8.6: Time history of dropping velocities (top) and fluid forces (bottom) for different initial velocities

As shown in Figure 8.6, the dropping velocity decreases more rapidly with a larger initial value. The three curves representing the velocities appear to be parallel to each other at the later stage, which means that deceleration is almost the same. For the fluid force, larger initial velocity generates higher fluid force at the early stage and a larger peak value. By contrast, the force values become almost the same at the later stage and this results in a same deceleration as observed in top figure of Figure 8.6.

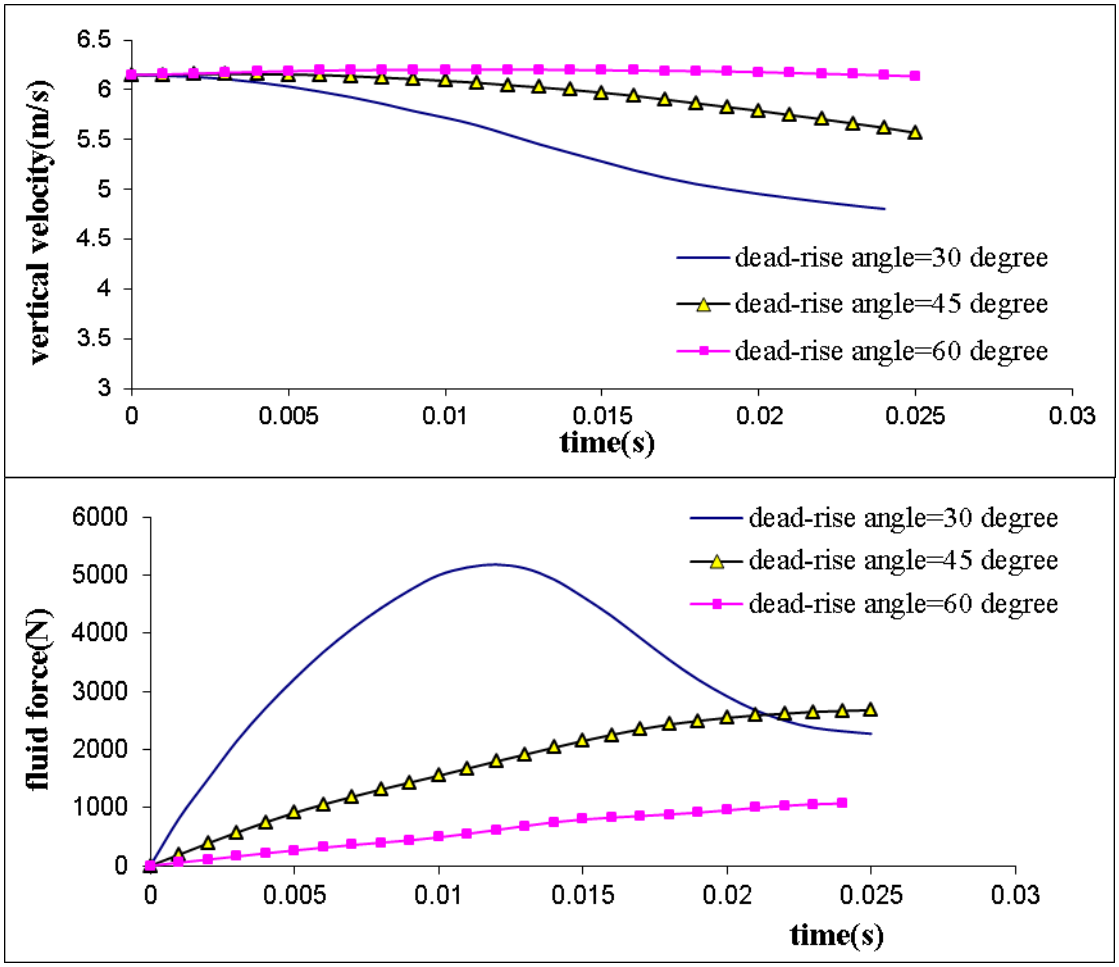


Figure 8.7: Time history of dropping velocities (top) and fluid forces (bottom) for different dead-rise angles

From Figure 8.7 we can see that fluid force acting on a wedge is smaller for the wedge with a larger dead rise angle and consequently the velocity decreases much slower. This is because the vertical force on the wedge is a projection of pressure from the fluid. This projection is based on the cosine of the dead rise angle. A larger dead rise angle will lead to a smaller value.

8.2 Dam breaking flow with a spring supported wall

In practice, large fluid impact force on the solid may result in movement of structure. It is interesting to know how the fluid force on the solid is affected due to the movement of solid. A spring supported rigid wall as shown in Figure 8.8 is selected as

an example to investigate the influence of solid movement on the fluid interaction force.

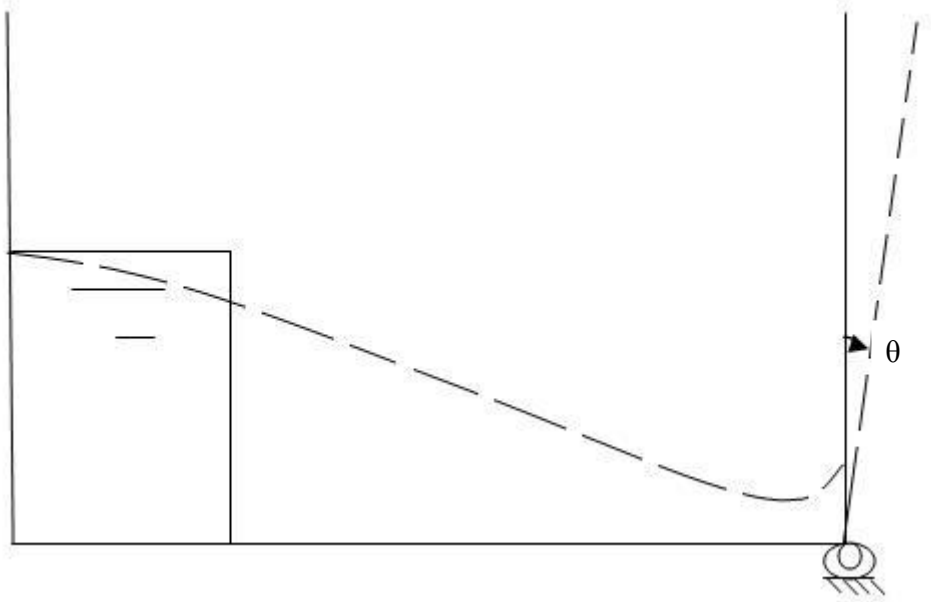


Figure 8.8: Spring supported rigid wall

In the simulation 120 x 60 particles with initial spacing of 0.01m are used for the inner fluid particles and for the spacing of boundary particle is 0.005m. All the settings in this case are the same as those used in the previous dam breaking case as discussed in section 7.3 (water column is 0.6m high and 1.2m wide, the wall is 3.22m long and 2m high). The only difference is that the right wall is attached to a spring which allows a limited rotational for the structure. The right wall stays stationary until the arrival of the fluid. It then starts to oscillate because of the combined effect of the fluid pressure and the restoring force of the spring. For a 3 second simulation the total CPU time is 5586.2s with time step size of 0.0001s, which is more than the CPU time of 1989.8s taken for the fixed wall.

The right rigid wall is only allowed to rotate and the following equation is used as the governing equation for the rotation (Xing, et al. 2003).

$$I\ddot{\theta} - MgH\sin\theta + K\theta = M_f \qquad (8.2.1)$$

where $I = 4/3MH^2$ is the moment of inertia, M is the mass with values of 1kg, H is the distance between the centre of mass of the wall and the pivotal point of the rotation with value of 1m, K is the stiffness constant of the spring of value of 20 and M_f is the moment produced by the fluid pressure. In the simulation, the moment M_f is calculated from the fluid pressure as:

$$M_f = \sum_{\mathrm{b}}^{\mathrm{N}} P_{\mathrm{b}} l \cos\theta dl \qquad (8.2.2)$$

Here, b is the solid particle in contact with water; P_b is the pressure at particle b, l is the distance from spring to particle b and dl is the particle spacing and θ is the spring angle.

Fluid flow motion with the fixed wall and rotational wall are depicted in Figures 8.9 and 8.10.

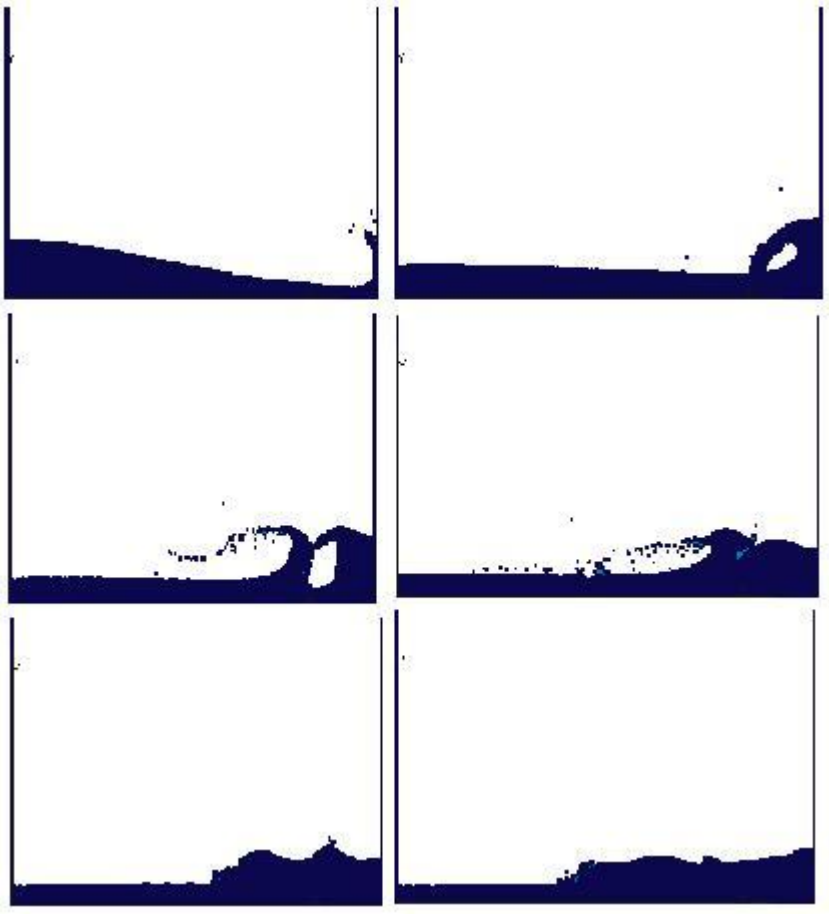


Figure 8.9: Motion of fluid pattern with fixed wall at t=0.8s, 1.4s, 1.7s, 1.9s, 2.2s and 2.45s

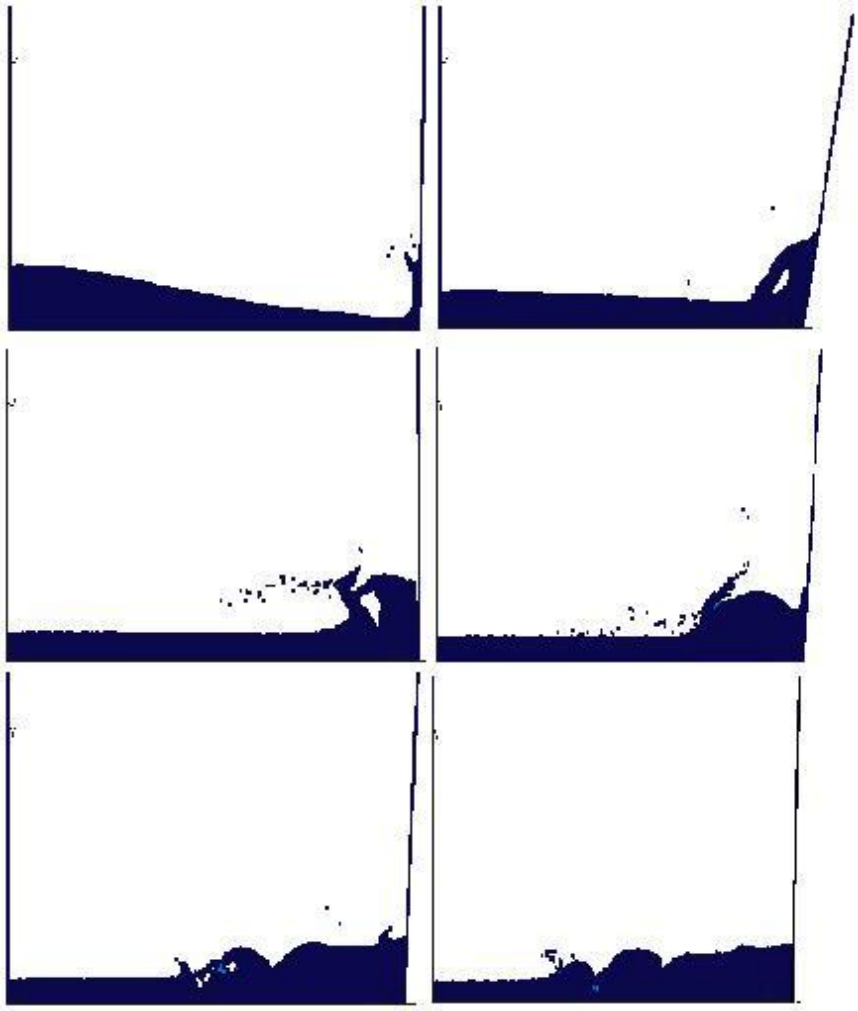


Figure 8.10: Motion of fluid pattern with spring supported wall at t=0.8s, 1.4s, 1.7s, 1.9s, 2.2s and 2.45s

Comparing Figure 8.9 and Figure 8.10 we can see that before 1.4s the motions of the water are similar for the two cases, but stronger waves are generated afterwards because of the rotation of the dam.

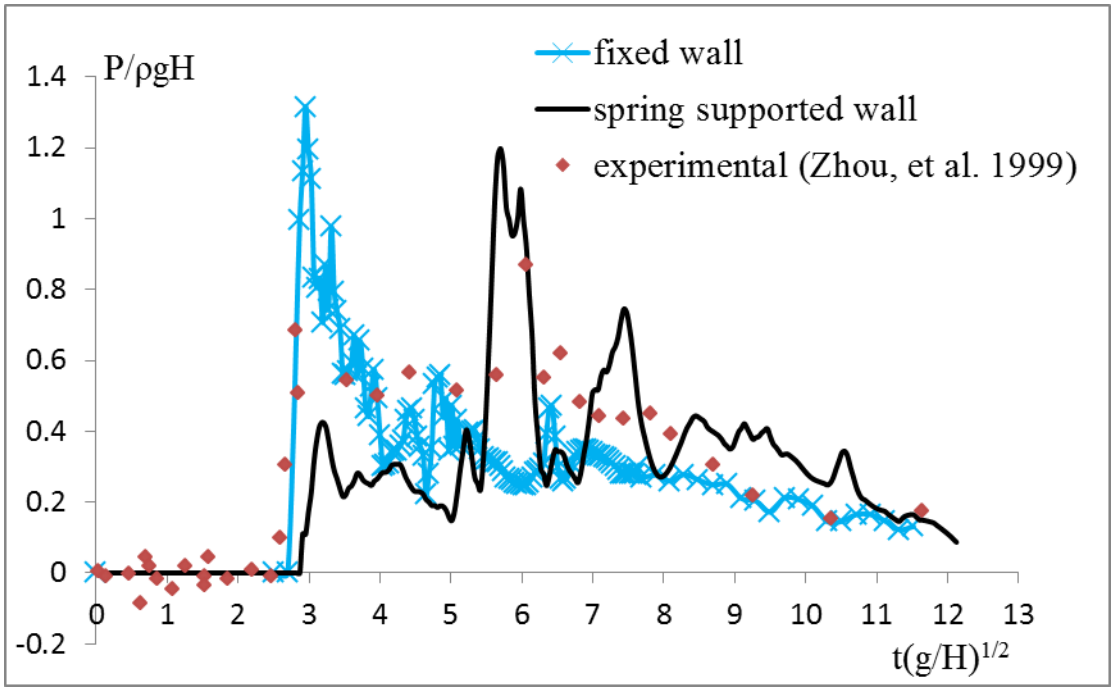


Figure 8.11: Impact pressure against downstream wall at point (3.22m, 0.16m) which is the initial coordinate for rotational wall

It is shown is Figure 8.11, before the second peak pressure values obtained in the situation when the dam can rotate are smaller. At the time around the second peak, the spring supported dam provides higher pressure values than the fixed case because after the first peak the fluid force on the solid decreases but the dam continues to rotate as a result of inertia. The spring keeps storing energy during this period. When the fluid pressure starts to increase again the spring achieves the maximum angle and then it forces the dam to return back, resulting in a stronger second peak. The values of these repeated peaks will be different with different spring stiffness. If the stiffness is large enough the wall could be treated as fixed. The vibration of the wall depends on the natural frequency of the spring (equals 0.71 in this case) as well as the impacting force. In fluid structure interaction cases, vibration and impacting force influence each other at the same time.

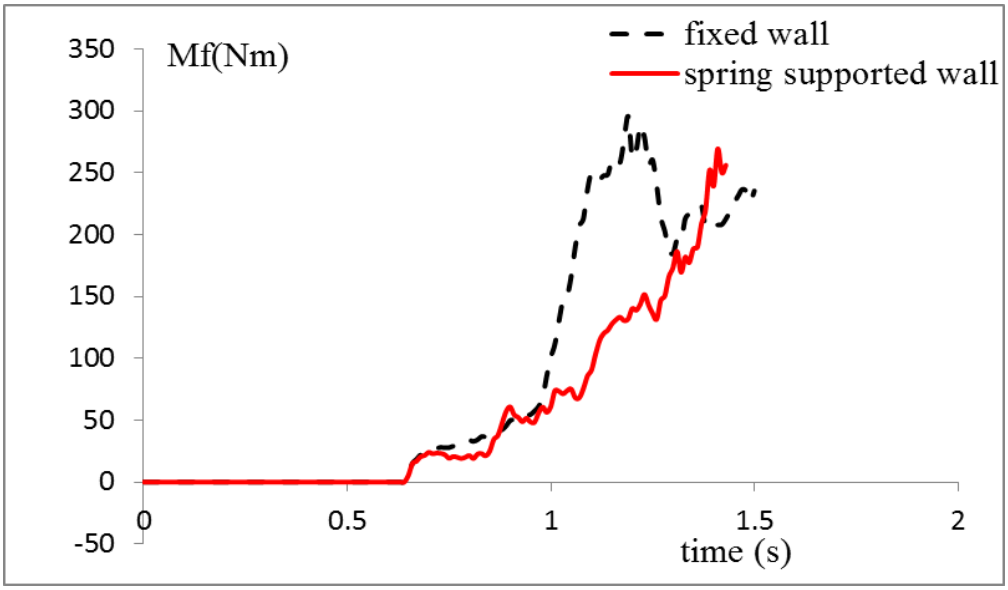


Figure 8.12: Early stage of impact fluid force momentum on the dam

Here, when the wall moves with the flow, the pressure peak is delayed as shown in Figure 8.11 but when the wall moves against the flow pressure peak is sped up as shown in Figure 8.12.

8.3 Summary

ISPH method is applied to fluid solid interaction in 2 dimensional cases. The two boundary treatments, i.e. repulsive force and denser wall particles on the boundary, discussed in the previous chapter are applied. The examples of wedge dropping and spring supported dam are used to demonstrate the performance of the method. The results show close agreement with experimental data. The influence of the parameters of wedge dropping problem including initial velocities, masses and dead rise angles are analysed. It was found that a larger dead rise angle or a larger weight would result in a slower deceleration; a larger mass or a larger initial velocity or a smaller dead rise angle would result in a higher vertical force. The flow involving spring supported wall is compared with that of a fixed wall and it shows that the movement of the all will delay the pressure peak.

Chapter 9 Application to air-water two phase flow

In marine and coastal engineering fields, violent fluid-structure interactions can lead to air entrapment. Simplifying these problems as incompressible fluid interacting with a solid will not capture the true physical nature of the problem. The air phase may have a large influence on the water flow evolution and subsequently on loads on structures. Therefore, the application of the SPH method on incompressible fluid is extended to two phase flows involving air and water. The advantage of SPH for multi-phase flow is that each phase of fluid follows its Lagrangian motion therefore the material interface is represented in a self-adaptively manner without the need for complex interface-capturing or front-tracking algorithm (Adami, et al. 2010). To model the compressible property of air and the incompressibility of water, a new method is proposed for air-water two phase flow simulation. These two different fluid phases are treated separately within the same time step. Air is solved using WCSPH and water is solved using ISPH with constant density. The time stepping algorithm is shown in Figure 9.1. No special treatment is required on the interface. The SPH formulations for multi-phase flow were given in Chapter 4. The standard SPH formulation for density and pressure gradient derived based on the assumption of continuous density of the material cannot be used for multi-phase flows especially for the cases when the density difference is large.

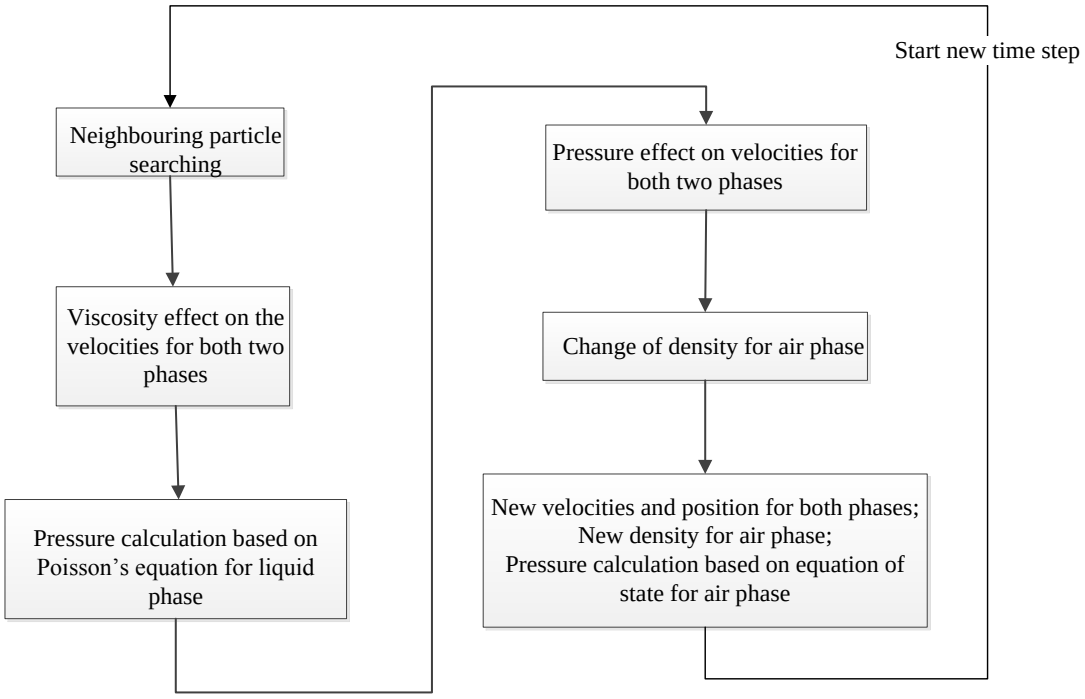


Figure 9.1: Time stepping algorithm for two phase flow using combined ISPH-WCSPH methods

In this chapter, the dam breaking flow with air-water two phase fluids is used to demonstrate the performance of the proposed algorithm. Results obtained from single phase flow and air-water two phase flow simulations are compared. Two typical multi-phase flow examples, i.e. rising air bubble in water and Rayleigh-Taylor instability are investigated to test the proposed method.

9.1 Air-water two phase dam breaking

The dam breaking case which has been studied in section 6.2 is simulated in this section taking into consideration of air-water two phase flow and the results are compared with the previous single phase case as shown in Figure 9.2.

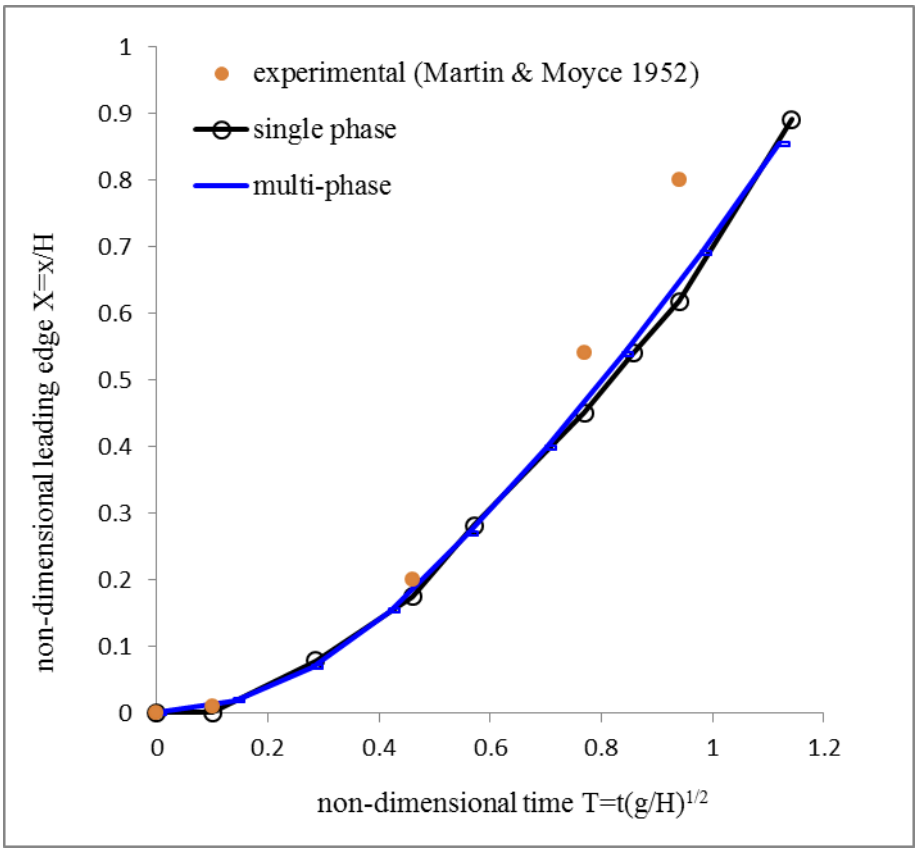


Figure 9.2: Position of the leading edge

The flow patterns of the fluid motion at different time instants are shown in Figure 9.3

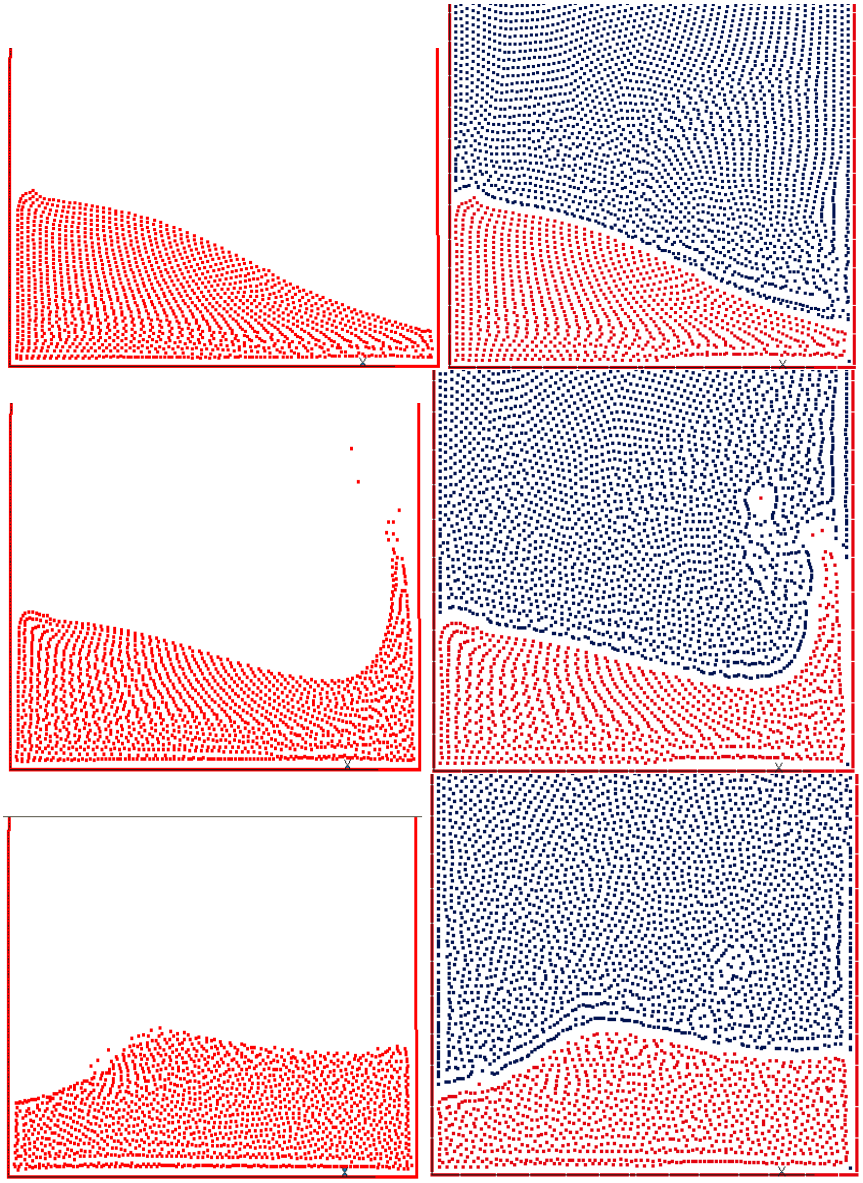


Figure 9.3: Fluid motion of dam-break in single phase case and multi-phase case at time 0.13s, 0.2s and 0.5s

From Figure 9.3 it is clear that in the dam breaking case, there is no significant difference on the fluid motion whether the effect from air is considered. This indicates that the velocity in a single phase case and a multi-phase case should be similar. A different initial air density is considered to simply test the behaviour of the algorithm. According to common experience, increasing the density of air should slow down the movement of water. When the initial air density is set to equal to that of the water, the whole fluid system is like a single phase fluid so that the water should stay stationary.

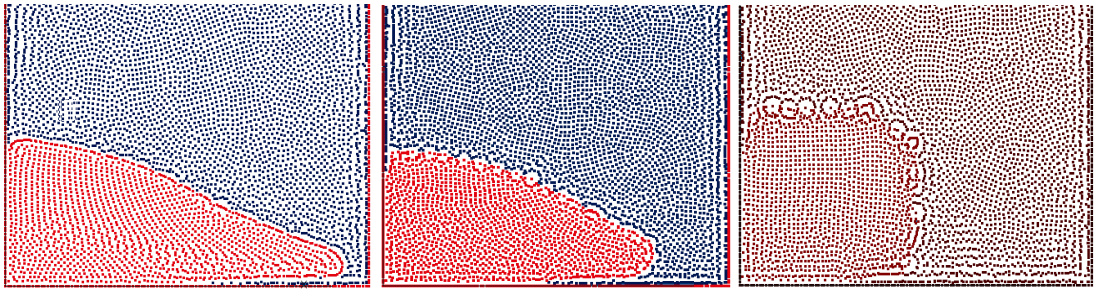


Figure 9.4: Multi-phase dam-break with air density increasing from 10 (kg/m^2) to 100 (kg/m^2) and 1000 (kg/m^2) at time 0.13s

Comparing Figure 9.4 with Figure 9.3, it is obvious that as expected the water flow is influenced by the density of the air. Increasing the air density will slow down the water movement. In the case when air and water have the same density, the water will hardly move. But because of the compressibility of air, the water column is deformed slightly under the pressure. This is consistent with our practical experience, which gives a certain qualitative validation of the proposed new approach.

To further prove the applicability of the proposed method for multi-phase flows, rising air bubble in water and Rayleigh-Taylor instability problems are simulated. Both the explicit and implicit solution approaches are used for the ISPH method.

9.2 Rising bubble

The model of the rising bubble example is sketched in Figure 9.5: a circular air bubble is free to rise through the initially stationary water. The number of particles used in this simulation is 60 x 100, with a particle spacing *dl=0.01*. The proposed method which combines ISPH and WCSPH methods is applied. And also WCSPH method is used for both fluids as a comparison. Repulsive force and denser wall particles are used on the boundary. The density ratio specified is around 0.001. Here the subscript a stands for air and w stands for water; the parameter used for the equation of state is $\gamma = 1.4$ for air. $\Delta t = 0.001$ for ISPH-WCSPH method and $\Delta t = 0.0001$ for WCSPH.

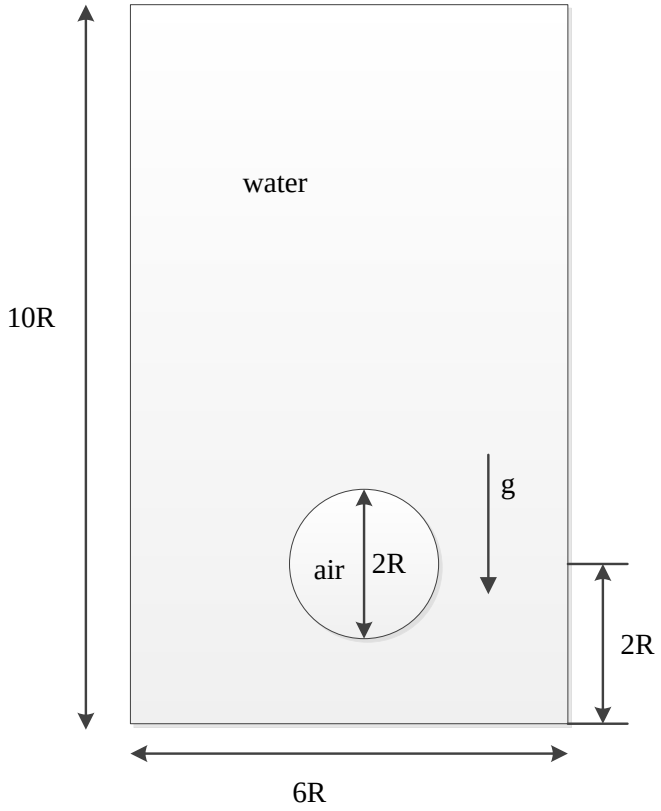


Figure 9.5: Sketch of the problem of air bubble rising in water

The results obtained by Sussman, et al. (1994) using different number of grids and different Bond numbers at 0.44s are displayed in Figures 9.6 and 9.7 as a reference. Here the Bond number is a dimensionless number used in the study of atomization involving bubbles and drops. By definition, Bond number equals to $(\rho - \rho')L^2 g / \sigma$, where ρ is the density of a bubble or drop, ρ' is the density of the surrounding medium, L is a characteristic dimension, g is the acceleration of gravity, and σ is the surface tension of the bubble or drop.

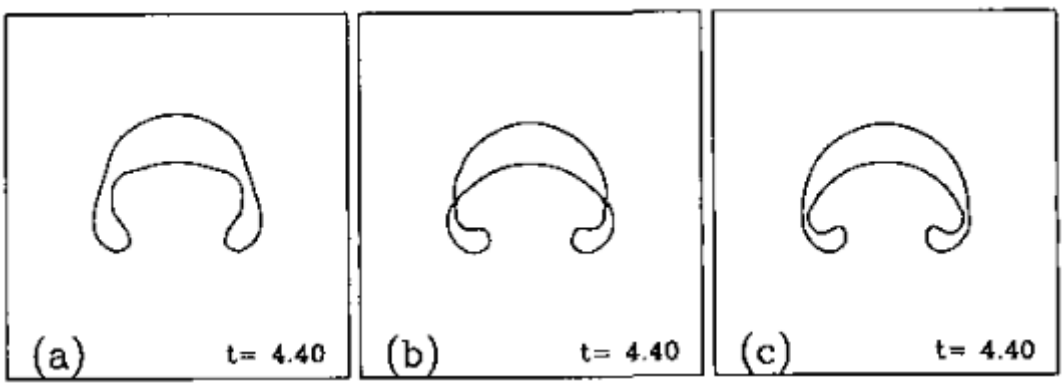


Figure 9.6: Convergence test for rising bubble with different grids form Sussman, et al.(1994) (t in the picture is non-dimension time defined as $t'(g/R)^{1/2}$, t' is the dimensional time)

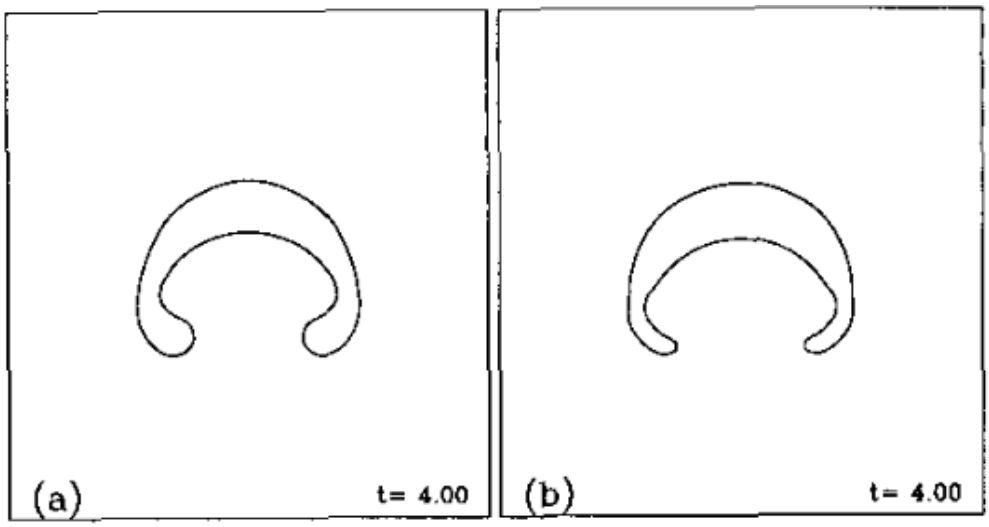


Figure 9.7: Results from Sussman, et al. (1994) of bubble rising with different Bond numbers: (a) Bond number 200.0 (b) bond number 25.0 (t in the picture is non-dimension time computed as $\Delta t(g/R)^{1/2} = 0.0001$)

It can be seen that the state of air bubble is influenced by various factors such as the numbers of grids. The exact state of the bubble is also sensitive to Reynolds number, surface tension, and density ratio, etc. In this project, three methods are applied: 1) ISPH-WCSPH with explicit solution process; 2) ISPH-WCSPH with implicit solution process and 3) WCSPH methods. The results are slightly different when different values of modelling parameters are used. A selection of results are shown in Figures 9.8 to 9.13 including the state of the bubble and the pressure distribution.

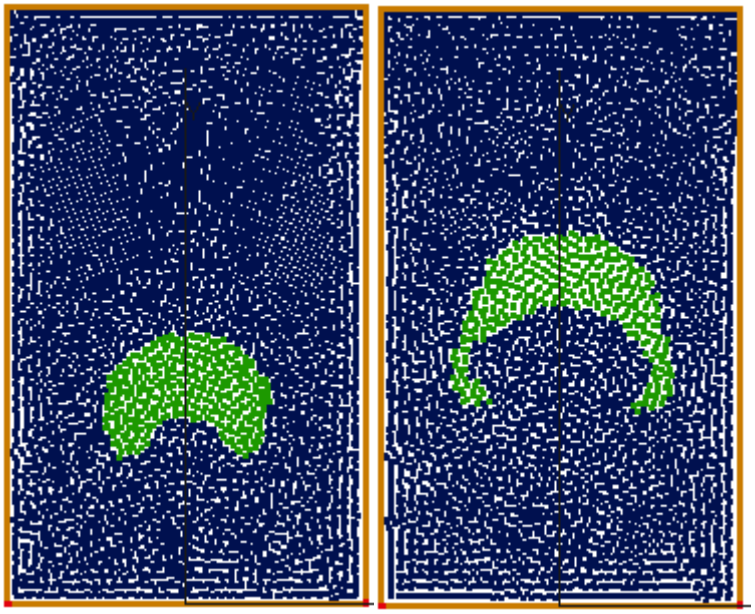


Figure 9.8: Motion at t=0.2 and t=0.45 with WCSPH method

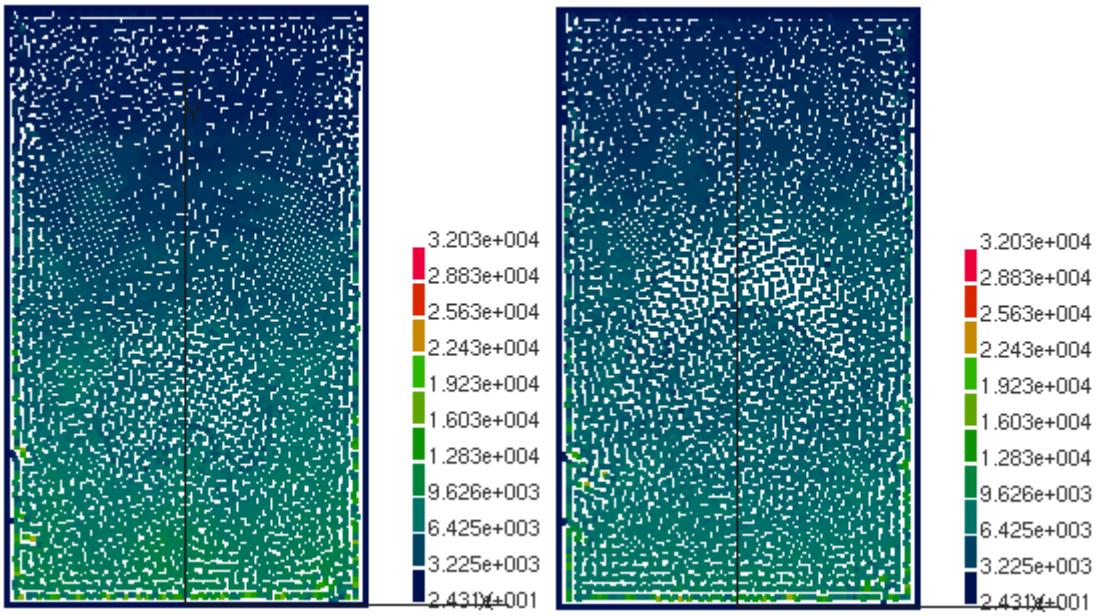


Figure 9.9: Pressure distribution at time of t=0.2 and t=0.45 with WCSPH method

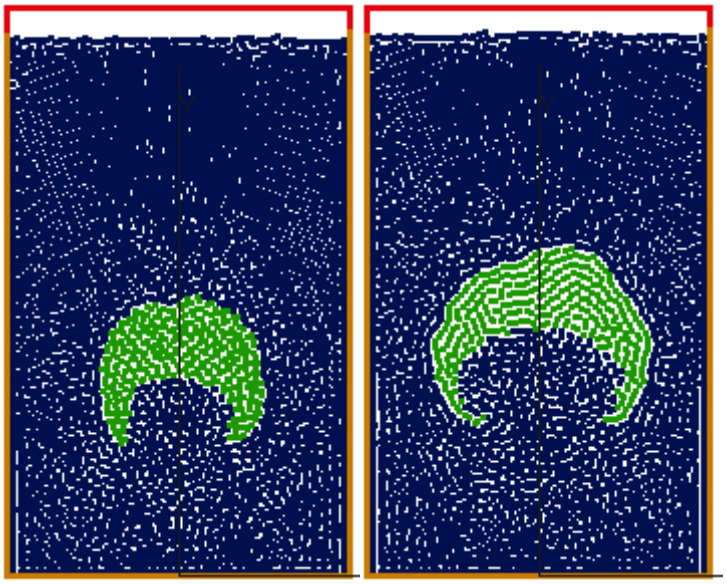


Figure 9.10: Motion at time of t=0.2 and t=0.45 using ISPH-WCSPH with explicit solution approach

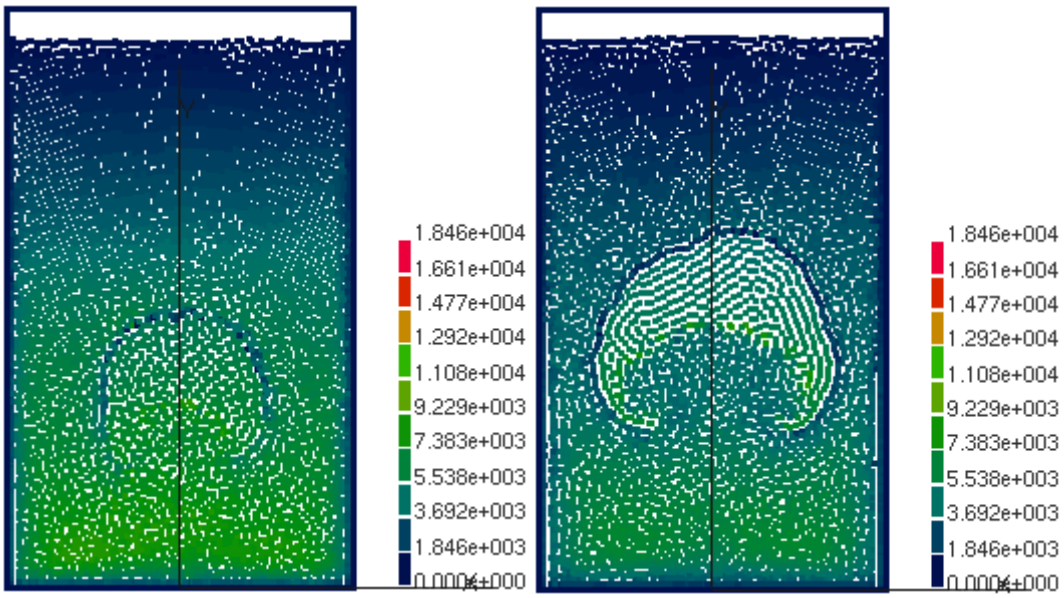


Figure 9.11: Pressure at time of t=0.2 and t=0.45 using ISPH-WCSPH with explicit solution approach

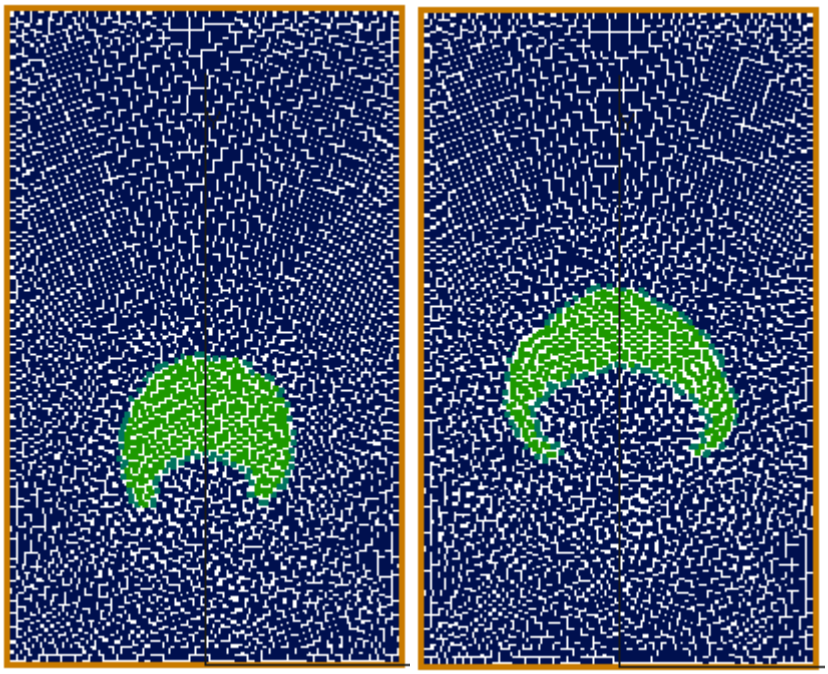


Figure 9.12: Motion at time of t=0.2 and t=0.45 using ISPH-WCSPH with implicit solution approach

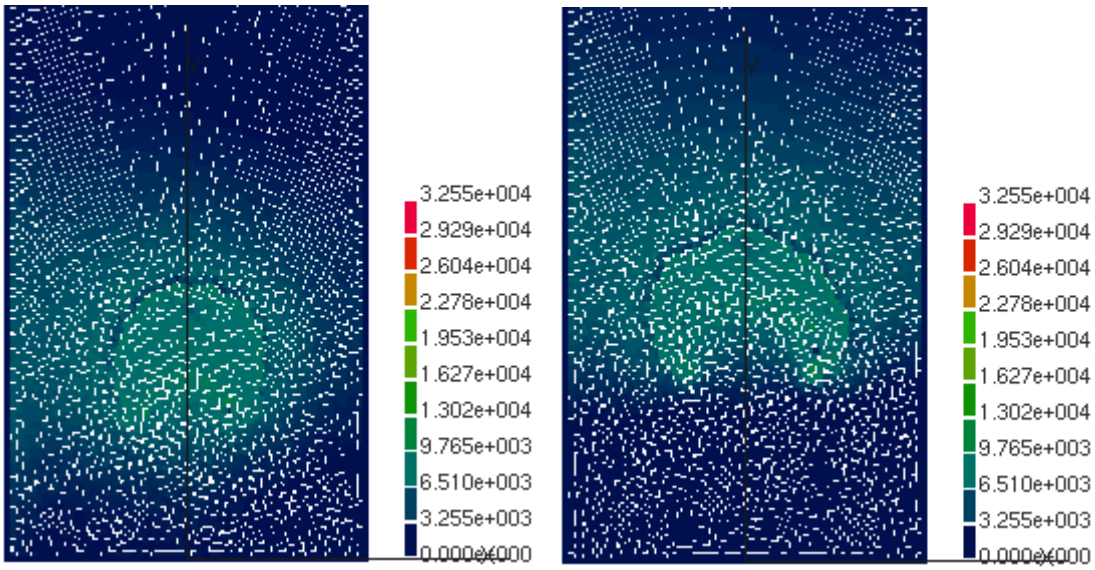


Figure 9.13: Pressure distribution at time of t=0.2 and t=0.45 using ISPH-WCSPH with implicit solution approach

The positions of bubble obtained using different methods are compared in the following figure

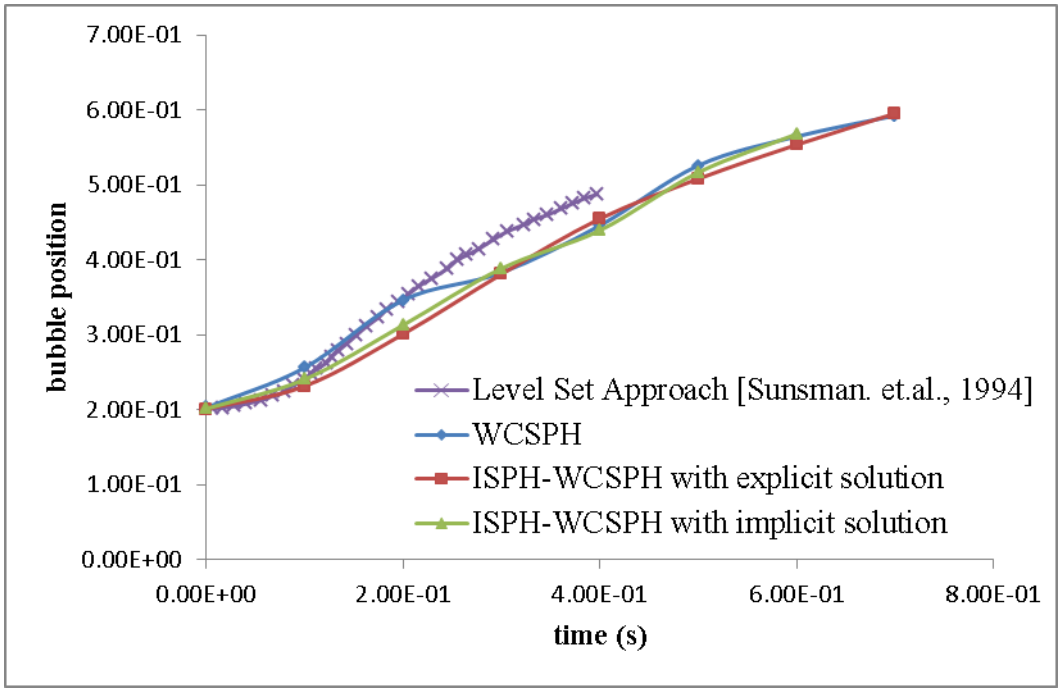


Figure 9.14: Bubble position comparation

From these figures it can be concluded that all three methods are able to provide reasonable predictions for the positions of the rising bubble. As time passes, the bubble rises, deforms and forms a horseshoe shape. Since the shape of the rising bubble is sensitive to many factors such as Bond number or the number of grids for the level set approach, the difference between SPH methods and the level set approach is expected as different parameters are used in those two different methods. It seems that the particles are distributed unevenly along a dragged interface. To improve the performance more particles can be used as shown in Figure 9.15 and 9.16. Also, a higher order of time stepping algorithm can be considered to improve the accuracy of SPH simulation. The combined ISPH and WCSPH method provides a better defined interface than the others. Pressure distribution is continuous in these two fluids except at the interface. It seems that there is a pressure jump at the interface especially when these two fluids are modelled with different methods. It indicates that there is a pressure difference produced by different methods but this difference is relatively small so that the whole system is still consistent.

method	Time-step size (s)	CPU time per time step(s)
WCSPH	0.00001	0.0219
explicit ISPH-WCSPH	0.0001	0.0739
implicit ISPH-WCSPH	0.0001	0.4874

Table 9.1 CPU time for rising bubble with different methods

The CPU time requirement for each method is given in table 9.1. Although the combined ISPH-WCSPH method can produce the smoothest results under the same conditions, it needs much more computation time than the other two methods. Generally, the combines ISPH-WCSPH method with the explicit solution approach has more potential to be widely used in the future.

As it is known that the accuracy of the SPH method can be improved by using more particles, it is interesting to see how the bubble will deform with more particles. Since WCSPH is the least time consuming approach and the result from the three methods are similar under the same model setting; only WCSPH is used in the following simulation to study the effect of particle spacing.

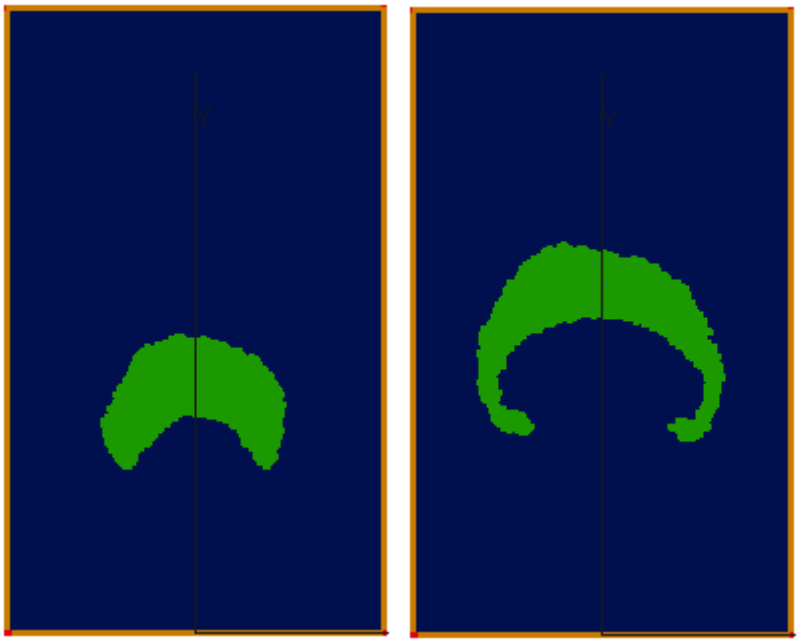


Figure 9.15: Motion at t=0.2 and t=0.45 with WCSPH method with particle spacing=0.005m (CPU time: 0.0445s per time step)

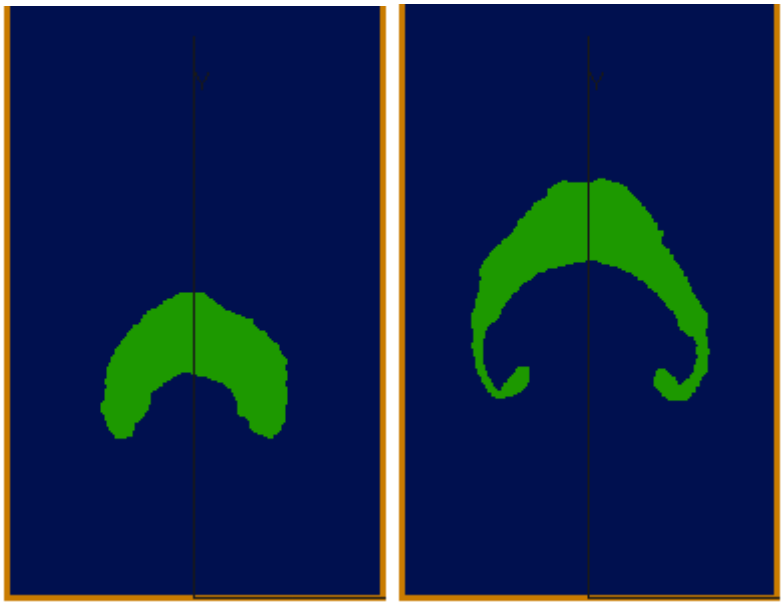


Figure 9.16: Motion at t=0.2 and t=0.45 with WCSPH method with particle spacing=0.001m (CPU time: 0.0923s per time step)

Comparing Figures 9.15 and 9.16, the shape of the deformation of the bubble with more particles is closer to the results obtained from the level set approach especially for the bottom side of the bubble. However, the CPU time is much higher due to the increased number of the particles.

9.3 Rayleigh-Taylor Instability

Rayleigh-Taylor instability is a classical testing case for the flow of two fluids of different densities. At the beginning, the heavier fluid is on top of the lighter fluid. The heavier fluid will descend and the lighter fluid will rise. Rayleigh-Taylor instability is considered in this section since this test case requires an accurate modelling of the interface between two different fluids. The model with a sinusoidal interface which is widely used is selected (Cummins & Rudman 1999; Hu & Adams 2007; Grenier & Touze 2008). The computational domain is rectangular as shown in Figure 9.17, a lighter fluid is filled in the lower part and a heavier fluid whose density is 1.8 times of the lighter fluid is filled above the interface located at $y = 1 - \sin(2\pi x)$. No slip boundary condition is applied. Five methods are used in this case: 1) ISPH with implicit solution; 2) ISPH with explicit solution; 3) ISPH-WCSPH with implicit solution; 4) ISPH-WCSPH with explicit solution and 5) WCSPH method. In the

equation of state, $\gamma = 7$ is used for both fluids. 60 x 120 particles are used in the simulation. The initial state of the fluid and pressure distribution is shown in Figure 9.17.

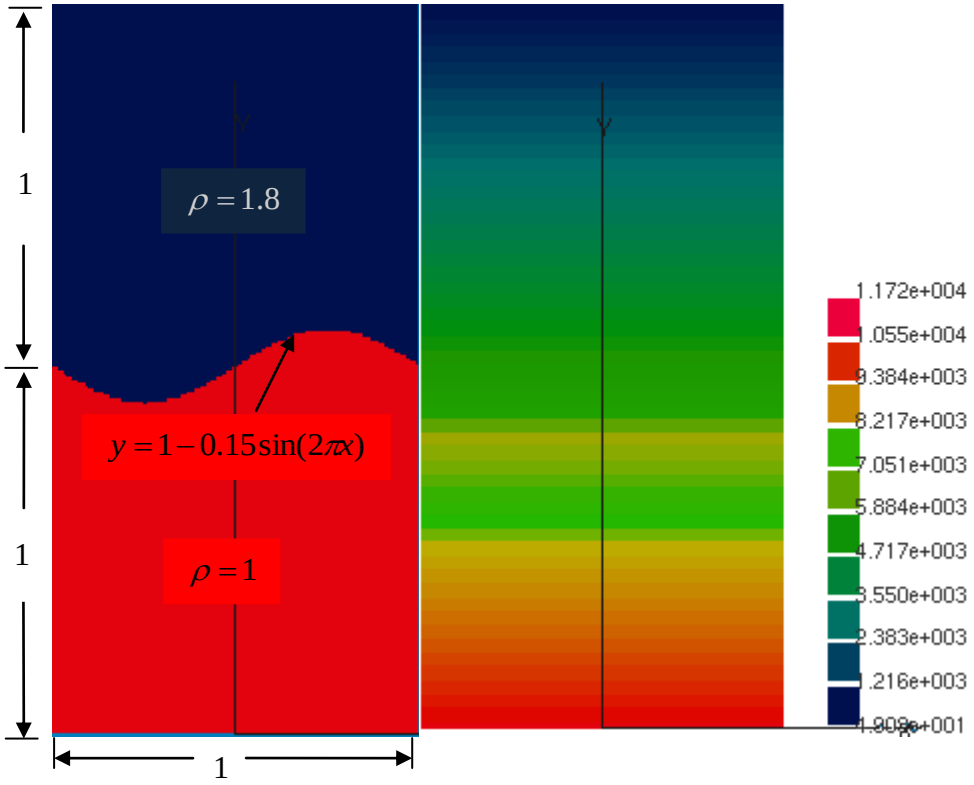


Figure 9.17: Initial state and pressure distribution of Rayleigh-Taylor instability

The result obtained using level-set method (Grenier & Touze 2008) is shown in Figure 9.18 as a reference.

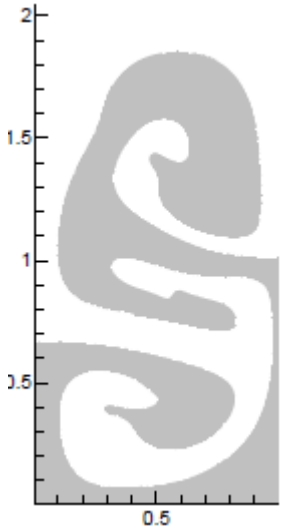


Figure 9.18: Rayleigh-Taylor instability problem simulated using Level-set method at time t=5 (Grenier & Touze 2008)

The interface positions and related pressure distribution at times t=3 and t=5 for each SPH method are shown in Figures 9.19 to 9. 28.

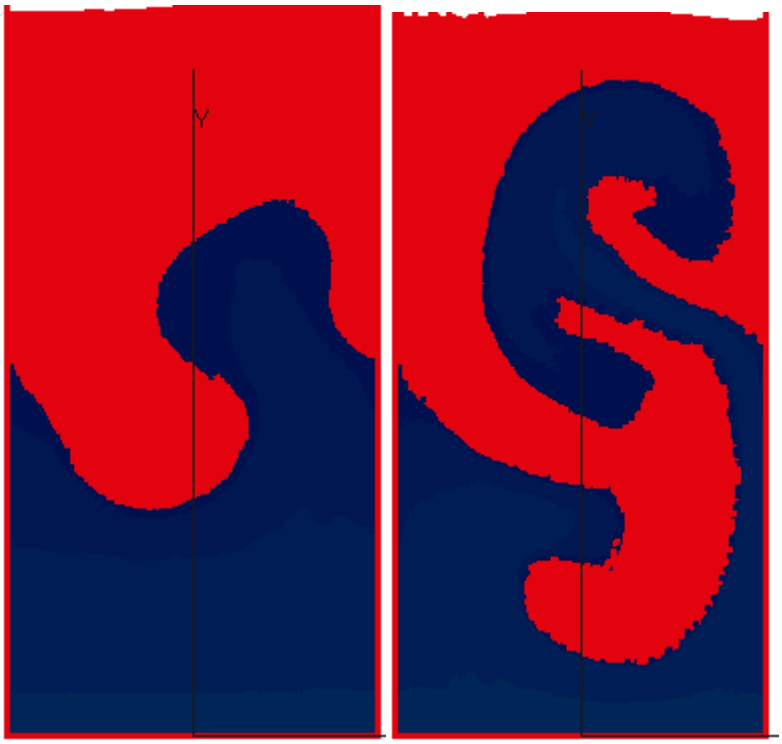


Figure 9.19: Flow pattern at time t=3 and t=5 using ISPH with explicit solution approach

The overall shape of the two-phase flow system at t=5 shown in Figure 9.19 is similar to Figure 9.18 and the interface is clearly defined.

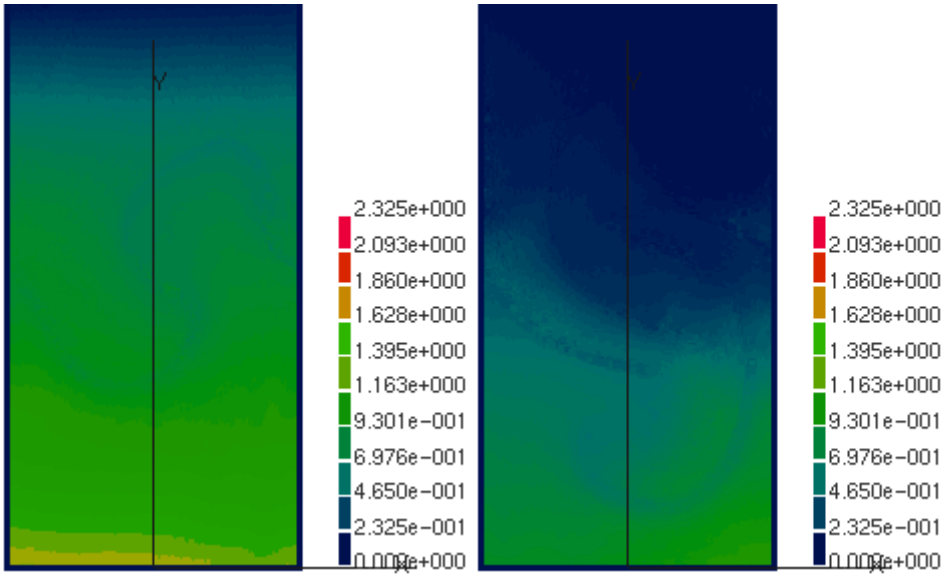


Figure 9.20: Pressure distribution at time t=3 and t=5 using ISPH with explicit solution approach

The pressure is distributed continuously as shown in Figure 9.20. As the legend indicates, a brighter colour represents a higher pressure. Hence, it is reasonable to see the brighter colour on the bottom of the fluid rather than on the top. At time t=5 the darker colour turns to cover more area than at time t=3. This means that the pressure of the system is reduced as the velocity of the fluid increases.

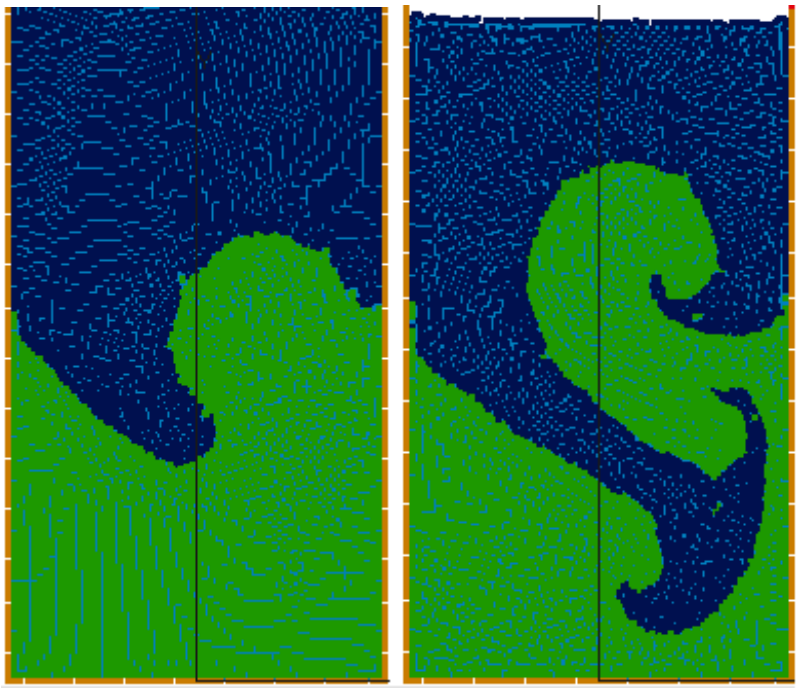


Figure 9.21: Motion at time of t=3 and t=5 using ISPH with implicit solution approach

From Figure 9.21 we can see that the heavier fluid moves down and the lighter fluid rises up and the interface is still clearly defined. However, comparing Figure 9.21 with Figure 9.19, it is seen that the volume of the heavier fluid sinking into the lighter fluid is reduced when ISPH method with implicit solution approach is used.

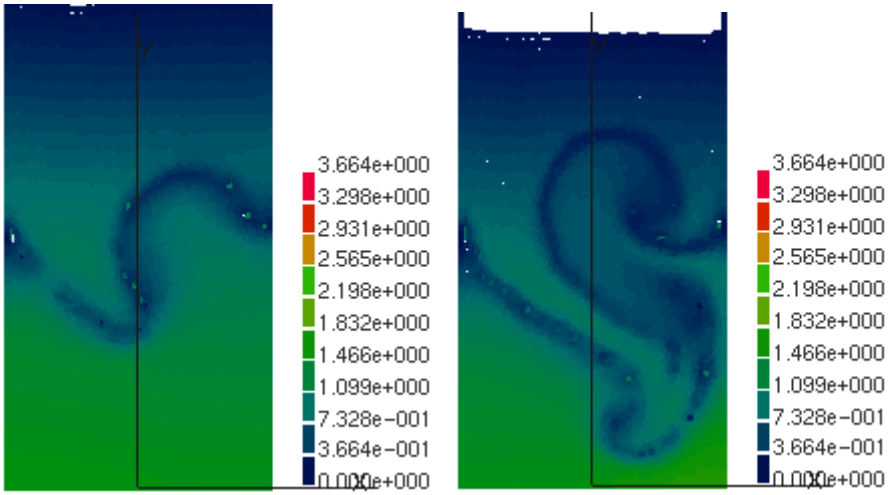


Figure 9.22: Pressure distribution contour at time t=3 and t=5 using ISPH with implicit solution approach

Comparing Figure 9.22 to Figure 9.20, it is noted that the pressure distribution obtained from explicit solution approach is different from the results obtained from the implicit solution approach. Except in the area near interface, the pressure distribution in the rest of the fluid seems only slightly changed from time t=3 to t=5 in Figure 9.22 and the overall height of the fluid is decreased. This may be related to free surface treatment as the implicit solution approach requires boundary values. In the cases when the space is fully filled with fluid no particles will be identified as free surface particles and hence no boundary values will be specified. Therefore, special treatment on the initial model setting is required to improve the performance of this approach.

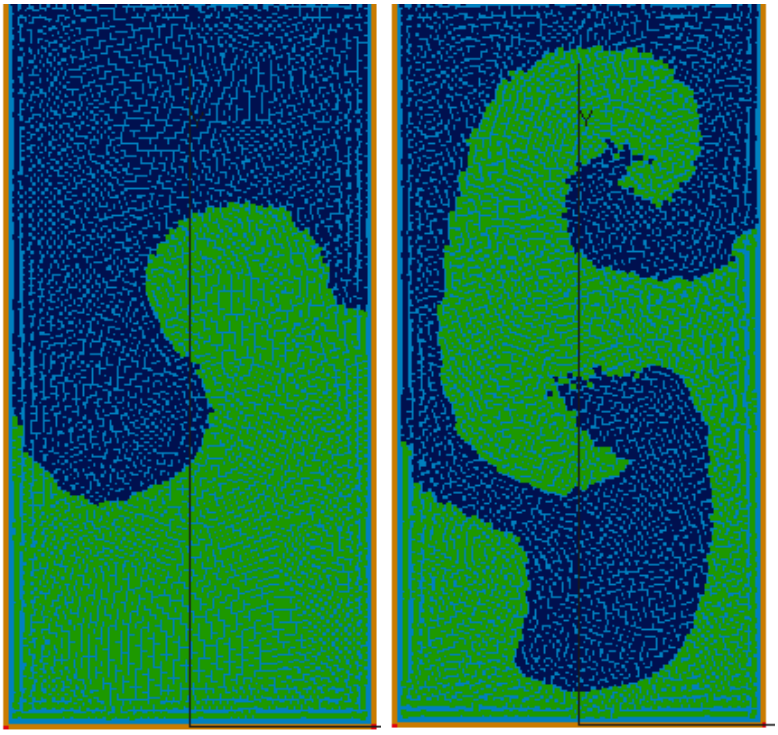


Figure 9.23: Motion at time of t=3 and t=5 using ISPH-WCSPH with explicit solution approach

The overall flow shape shown in Figure 9.23 is similar to Figure 9.18 although the volume of the incompressible fluid seems slightly larger.

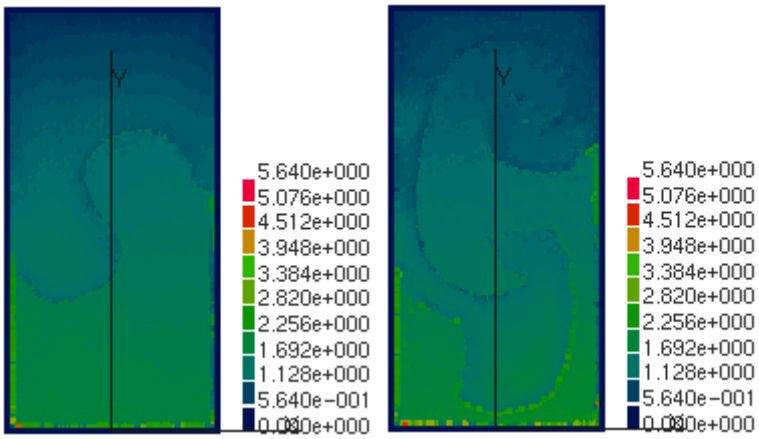


Figure 9.24: Pressure distribution contour at t=3 and t=5 using ISPH-WCSPH with explicit solution approach

Pressure distribution shown in Figure 9.24 is generally continuous. Pressure values of incompressible fluid seem slightly lower than the compressible fluid at the same level. The accuracy of pressure prediction still needs to be improved.

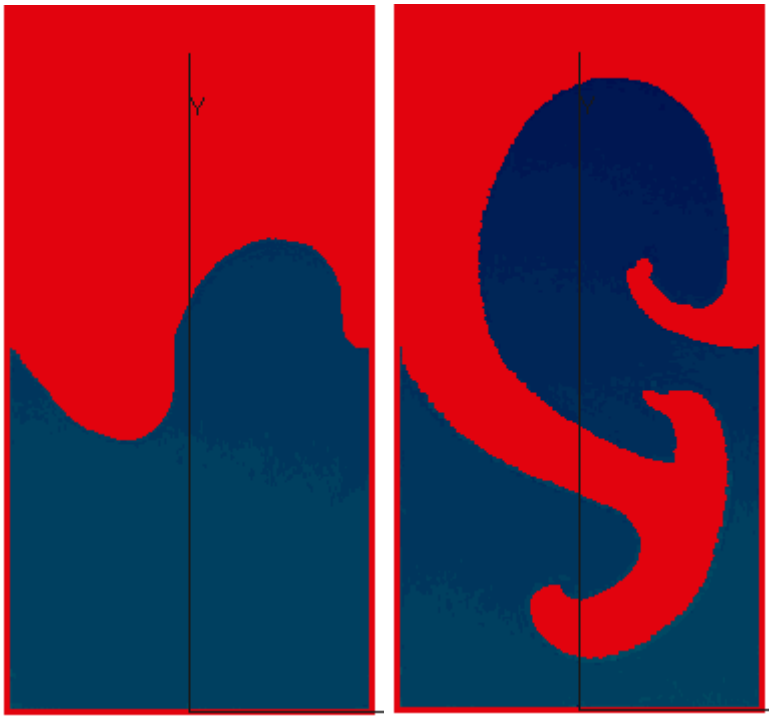


Figure 9.25: Motion at time of t=3 and t=5 using ISPH-WCSPH with implicit solution approach

Comparing Figure 9.25 and Figure 9.18, the shapes of the flow are similar and the interface is clearly defined. However, in contrast to the previous case in which explicit solution approach is used, the volume of the compressible fluid (lighter fluid) at time t=5 seems expanded slightly.

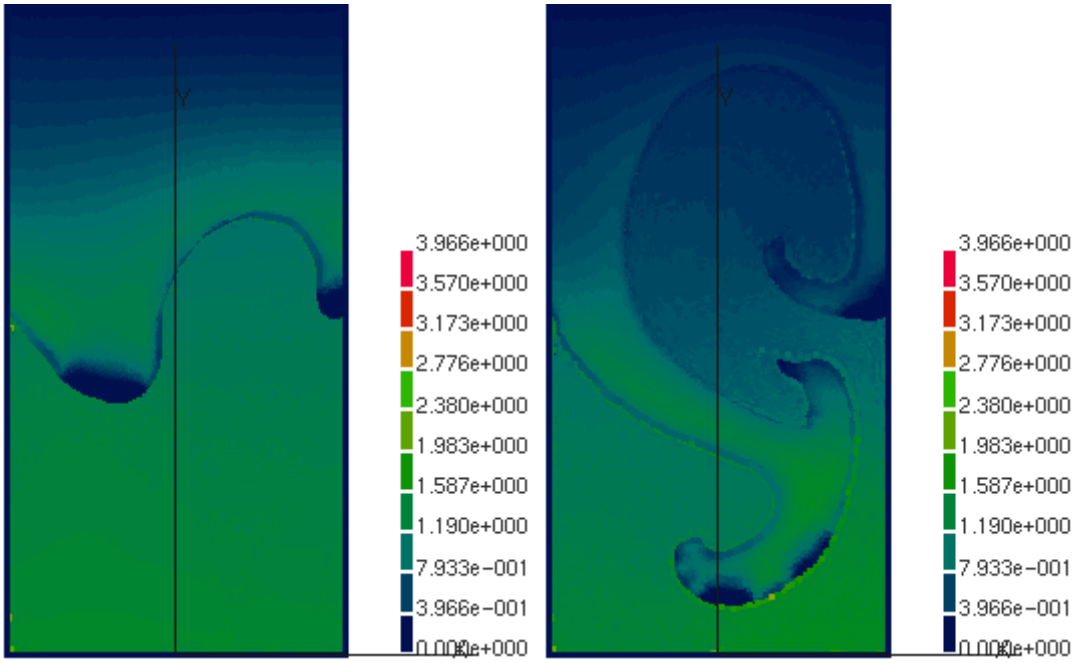


Figure 9.26: Pressure distribution at t=3 and t=5 using ISPH-WCSPH with implicit solution approach

The overall pressure distribution shown in Figure 9.26 is clearly continuous except in the area near the interface. Some lower pressure values are observed at the interface especially for the incompressible fluid at t=3. By contrast, pressure values of incompressible fluid are slightly higher than the compressible fluid at the same level at t=5. Slight difference of pressure values is acceptable because the values are obtained from two different algorithms.

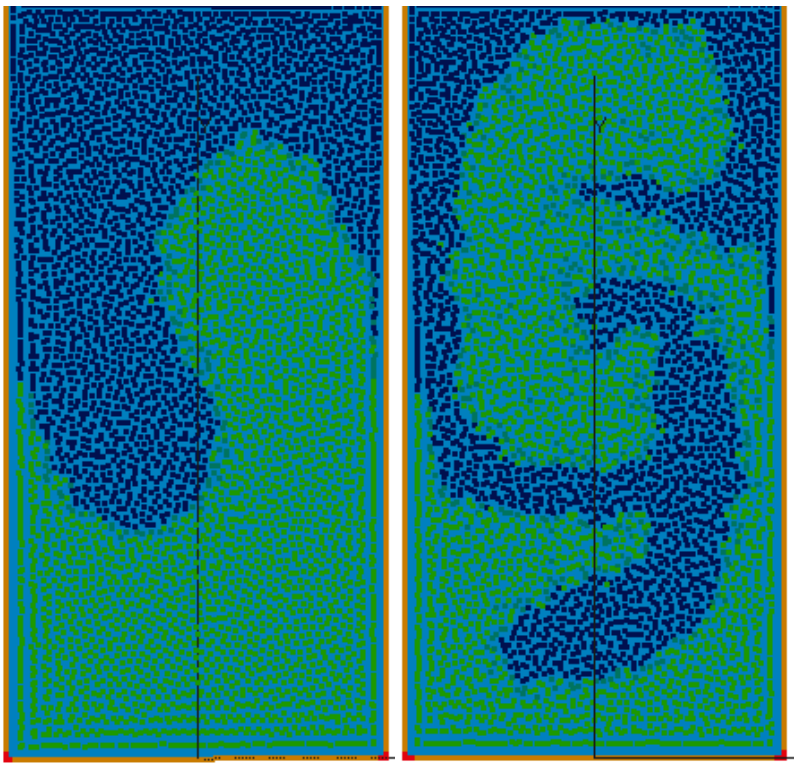


Figure 9.27: Motion at time of t=3 and t=5 using WCSPH

The shape of the flow shown in Figure 9.27 is reasonable as it is similar to Figure 9.18. However, the interface is less sharply defined compared to the results obtained from other methods.

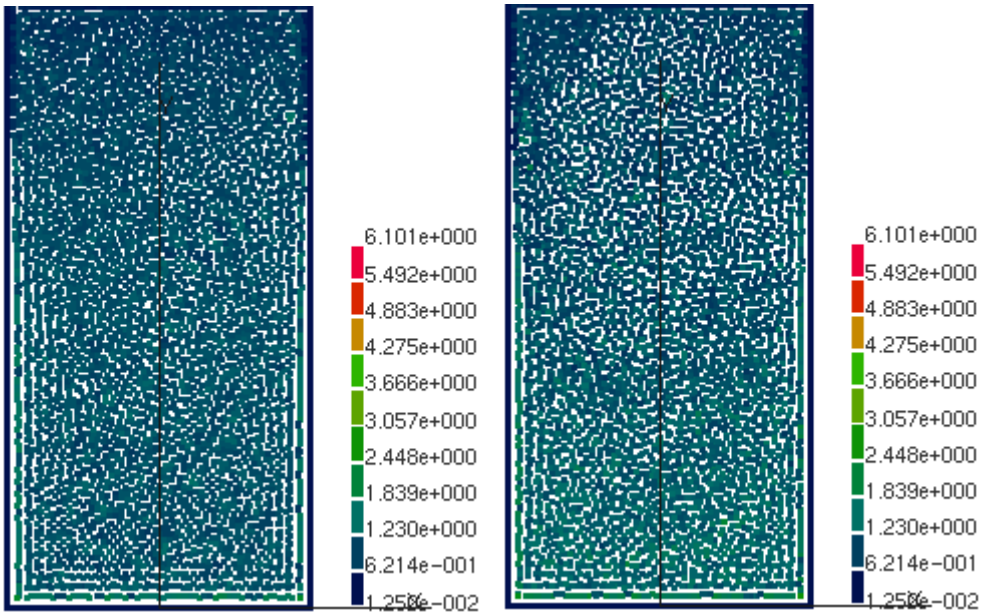


Figure 9.28: Pressure distribution contour at t=3 and t=5 using WCSPH

The pressure distribution is not clear in Figure 9.28. The pressure at different height cannot be read based on the colour.

From Figure 9.19 to Figure 9.28 and compared to Figure 9.18, although there are differences in the details of the results produced by these different methods, the overall motion of the fluid is reasonable. The interface is clearly identified from the simulation. Particle motions obtained from ISPH methods and ISPH-WCSPH with implicit solution approach are smoother than the others. Pressure distribution is clearly shown in ISPH method and ISPH-WCSPH method in contrast with WCSPH method where the pressure distribution is not clear.

method	Time-step size (s)	CPU time per step (s)
ISPH-explicit	0.0005	0.1889
ISPH-implicit	0.0005	0.4442
ISPH-WCSPH-explicit	0.0005	0.0264
ISPH-WCSPH-implicit	0.0005	0.6696
WCSPH	0.00005	0.0219

Table 9.2: CPU time for Rayleigh-Taylor instability simulation with different methods

It is clear from table 9.2 that the implicit solution approach for ISPH combined with WCSPH takes more computational time than the others. The ISPH method alone with the implicit solution approach requires the second most computational time. This implies that extra iteration is required for dealing with two phase flows. Besides, when ISPH combines with WCSPH, the pressure value needs to be transferred from the WCSPH solution in the Poisson's equation which leads to more iteration to reach the convergence. The interesting point is that the computational time is reduced as much as thirty times when WCSPH is combined with ISPH using explicit solution approach. This combination requires the least computational time. This indicates that the ISPH method with explicit solution approach has more potential in the future.

9.3 Discussion

A new method which combines ISPH and WCSPH methods for air-water two-phase flow is developed. For incompressible fluid, constant density is used, while for compressible fluid the change of density due to compressibility is calculated. The initial consideration for this method is to reduce computational time as well as to preserve accuracy. However, from the results it is clear that combining ISPH using the implicit solution approach with WCSPH method will increase the computational time because of extra iteration. By contrast, the computational time can be reduced significantly when WCSPH is combined with ISPH using the explicit solution approach.

Two testing examples: rising bubble in water and Rayleigh-Taylor instability problem have been simulated using the proposed methods. Although minor differences can be seen from the different methods, the fluid motion is well predicted by all methods used. Hence, it can be concluded that the new method developed is capable for simulating air water two-phase flows. However, the results can be further improved by using different parameters or a higher order of time stepping algorithm. In addition, certain correction methods such as XSPH can be considered. In dam breaking simulation, no significant change is observed when effect of air is considered.

In short, the proposed new method which combines ISPH and WCSPH methods has shown great potential for multi-phase flow simulations with large density ratios.

Chapter 10 Application to 3D fluid rigid body interactions

To obtain more realistic results for simulating a general fluid solid interaction problem, a 3D algorithm with ISPH method has been developed. The solid is assumed to be rigid so it can be considered as moving boundaries for fluid. Since repulsive force has been proved to be efficient for ISPH method with 2D examples it will be used for 3D simulations in this chapter. The advantage of this boundary treatment will be more important in 3D simulations of fluid-structure interaction problems as it requires the least number of particles on the boundaries compared to other boundary treatments. The algorithm can be extended to deformable solid interacting with fluid by using elastic or plastic theories for solid.

Due to the limitations of memory space and CPU time consumption only single phase flow is considered for fluid. Some examples including dam breaking and water entry of wedge which have been studied in the previous chapters are simulated again in 3D for comparison purpose. Finally, aircraft ditching and landing gear running on the wet runway are investigated.

10.1 3D dam breaking

The experimental model for dam breaking case study used in section 6.2, 7.1.and 9.1 will be adopted here. The only addition in the model setting is along the third direction there is a depeth of 1m. The initial particle spacing is 0.01m. Quartic kernel funciton is used, smoothing length is 1.33 times of the spacing. The patterns of the flow at time 0.005s and 0.007s are shown in Figure 10.1.

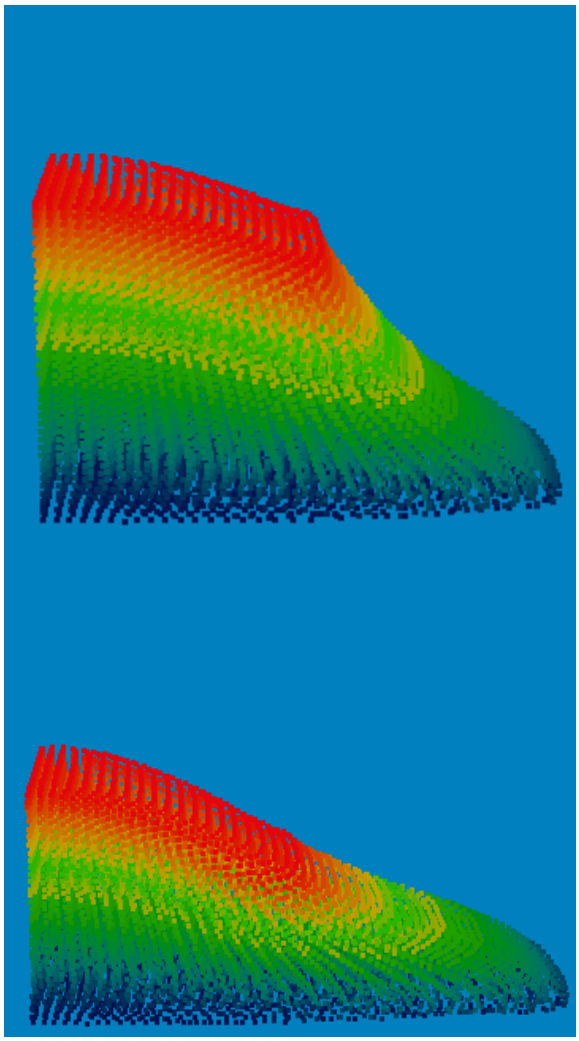


Figure 10.1: Fluid patterns at time 0.005s and 0.007s

A smooth flow pattern is shown in Figure 10.1 whearas in Figure 10. 2 the position of the leading edge of the fluid predicted using 3D SPH method is shown together with 2D results and experimental data.

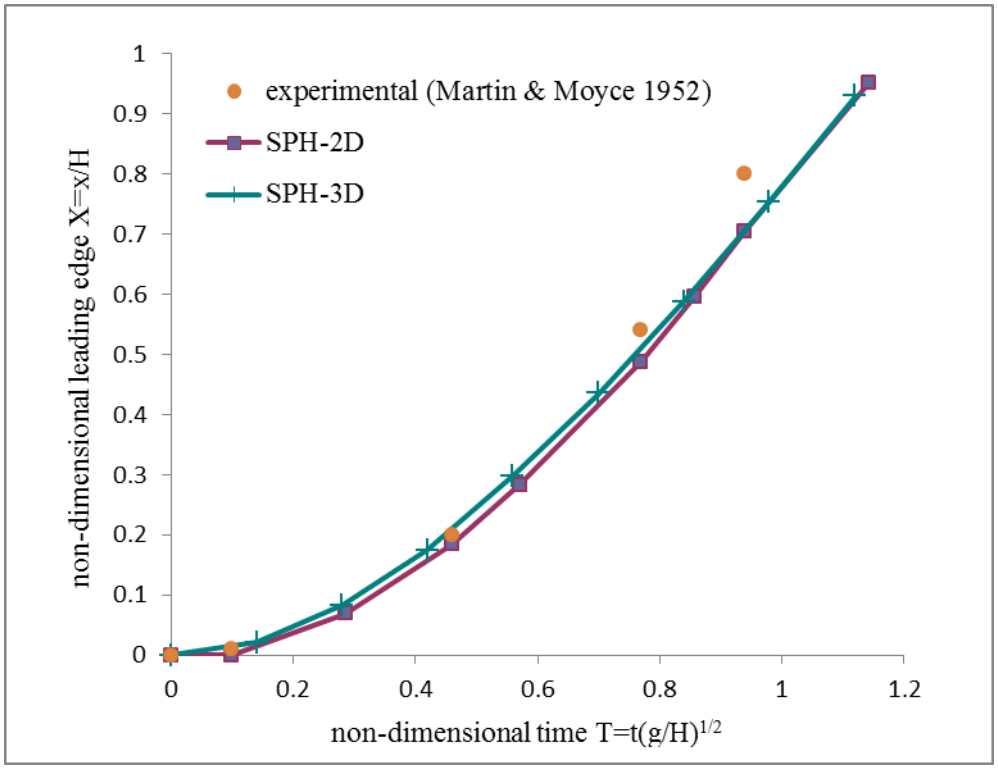


Figure 10.2: Position of leading edge of dam break

From Figure 10.2 it is clear that the results obtained from 3D simulation is closer to the experimental data compared to 2D case, although the difference is not significant since the flow is in fact largely 2D. But this gives a preliminary verification of the 3D algorithm. In the next section, the wedge dropping example will be studied in the 3D case to provide more evidence for the performance of the algorithm.

10.2 3D wedge dropping

The wedge dropping model is the same as the one used in section 8.1 with an additional consideration of the third direction. Particles are distributed uniformly with spacing of 0.01m and time step size is 0.0001s. The motion of wedge and wave pattern generated during dropping are shown below in Figure 10.3.

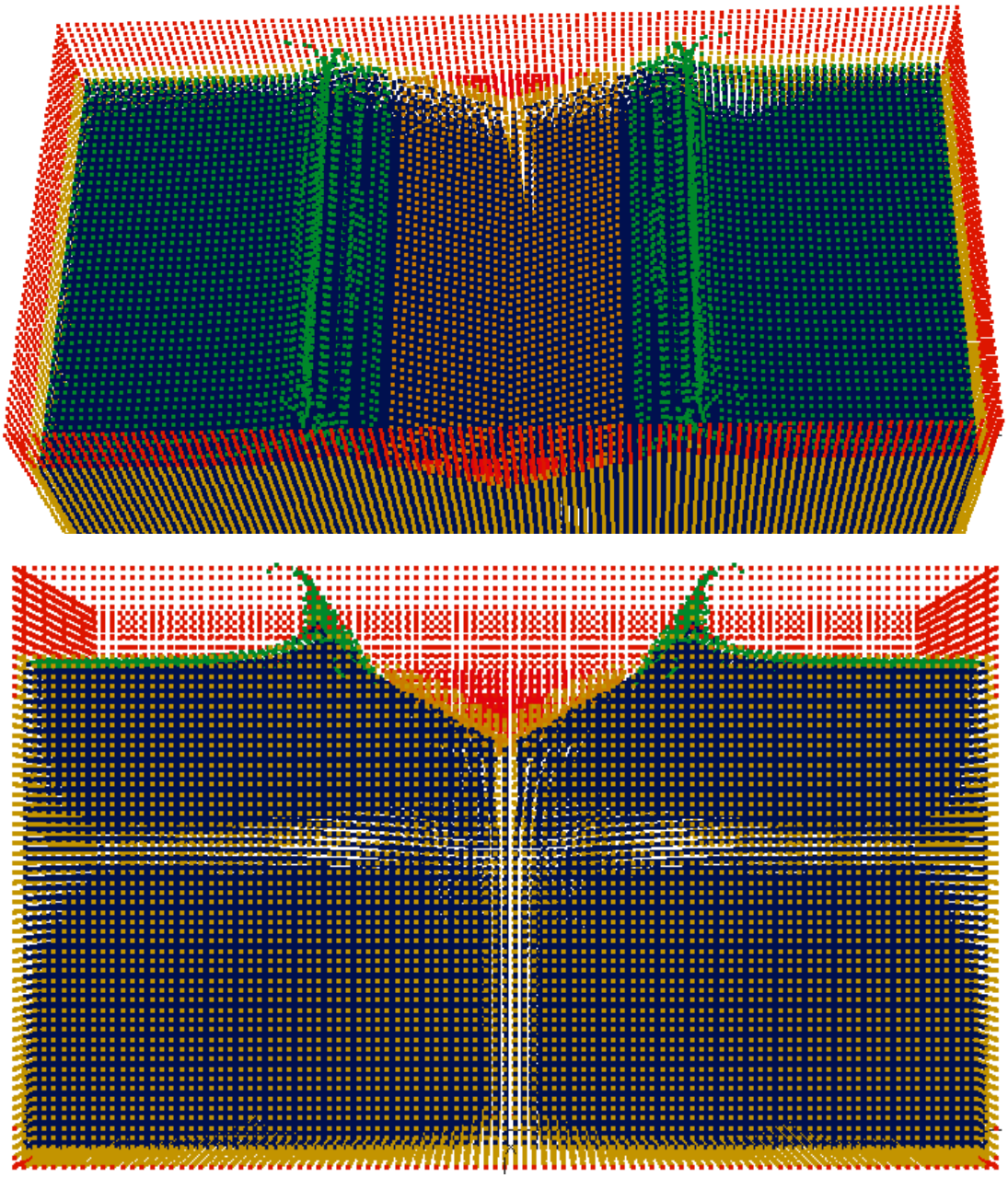


Figure 10.3: Motion of wedge dropping at time 0.035s viewing from above and front

From Figure 10.3 we can see that at the moment of the wedge entery, the surface of the water breaks into two parts symmetrically, with the water splash running along the edge of the wedge. This is the same as in 2D simulation. From the top view it is clear that the splashing water is forced up by the dropping wedge and then it plunges back into the rest of the water surface. The velocity and vertical force on the dropping wedge are compared with experimental data as well as the results from 2D simulation in Figures 10.4 and10.5.

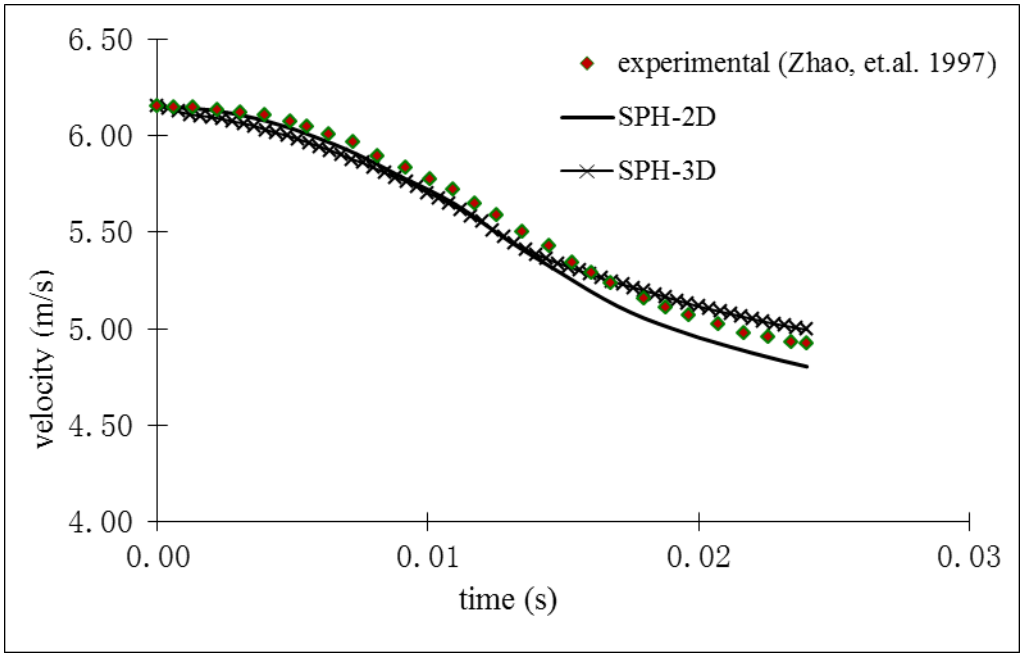


Figure 10.4: Velocity of wedge dropping

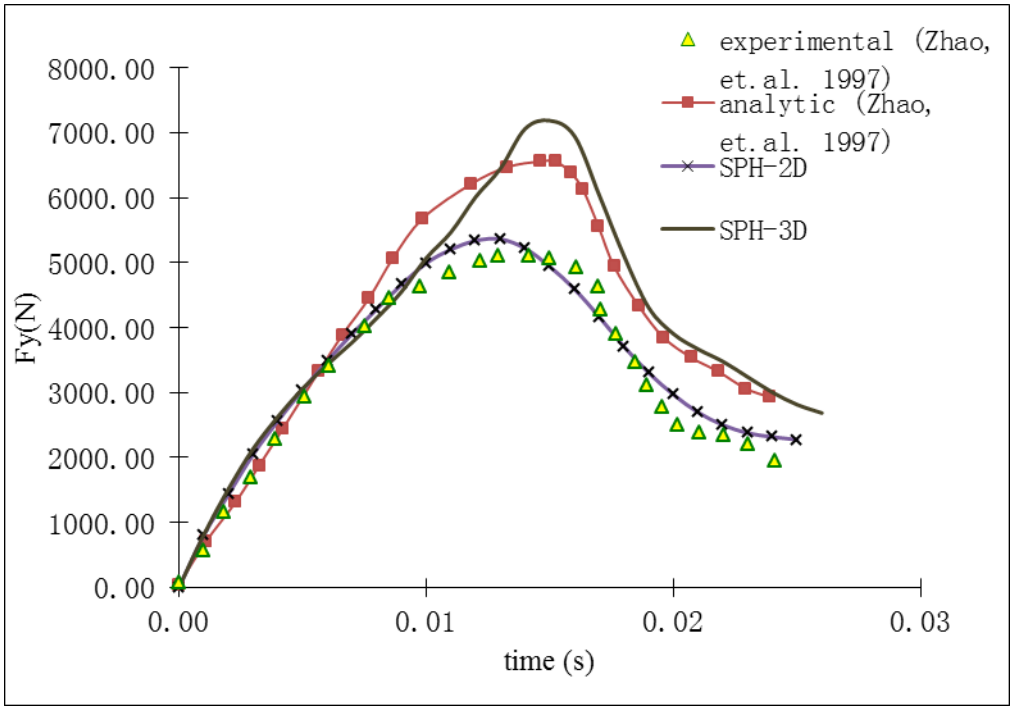


Figure 10.5: Vertical force profile for wedge dropping

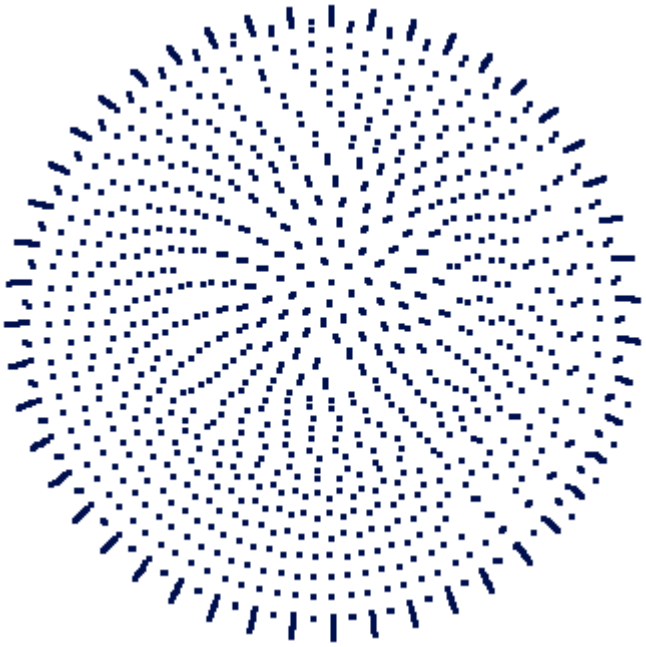
From Figure 10.4 it is clear that the velocities obtained from 3D simulation are closer to the experimental data. From Figure 10.5 it seems that the force on the wedge obtained from 2D simulation is closer to the experiment data whereas the 3D results are closer to the analytical results.

10.3 Water spray generated by landing gear

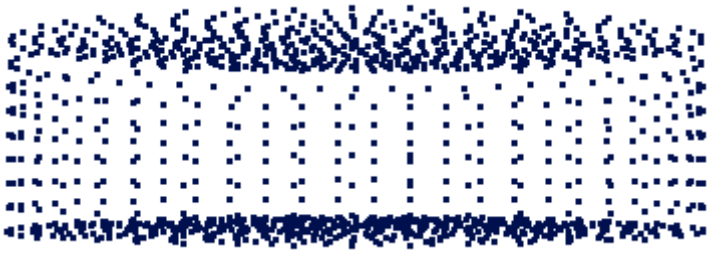
All aircraft designed to take-off or land on conventional runway must have ability to operate when the runway is wet. As it is known that a wet runway may affect the braking and manoeuvring capability and also it may lead to a reduction of the take-off acceleration. Water spray thrown up by the aircraft tyres could be ingested into the engine especially for those large multiengine aircraft with aft-fuselage-mounted turbojet engines. If sufficient water is ingested, a jet engine can experience compressor stalls or even flameout. This stall or flameout situation can be especially dangerous if it occurs when approaching the moment of take-off. Typically, the requirements of the commercial aircraft certification includes a section that the aircraft manufacturer demonstrate the capability to operate on a runway with one-half inch of standing water without experiencing any spray ingestion problems. Some aircraft have a configuration that is free of spray problems regardless of external conditions such as water depth and speed. Other aircraft have configurations that make spray ingestion a common problem over a wide range of conditions. These aircraft must fit with chained tyres or nose wheel spray deflectors. Numerous studies have been conducted to determine whether aircraft are susceptible to water spray ingestion, but they were usually carried out after construction of the aircraft. Although the design of aircraft and engine type and location are dependent on many variables, it is desirable to configure an aircraft and its engine in a geometry that eliminates the spray ingestion potential (Daugherty & Stubbs 1987).

Simulation of landing gear running on a wet runway is carried out in this section to study the water spray pattern.

As a simplified model, a single rotating wheel is used to represent the landing gear and the fuselage is not included in the model. In the simulation, the radius of the wheel is 0.15m and its width is 0.1m. The gear model is hollow inside to reduce the number of particles. The side view and top view of the model are shown in a) and b) of Figure 10.6.



a): side view of the gear model

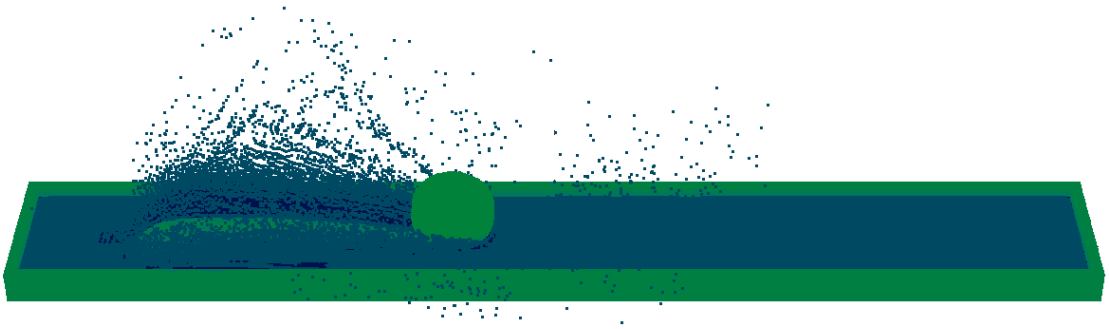


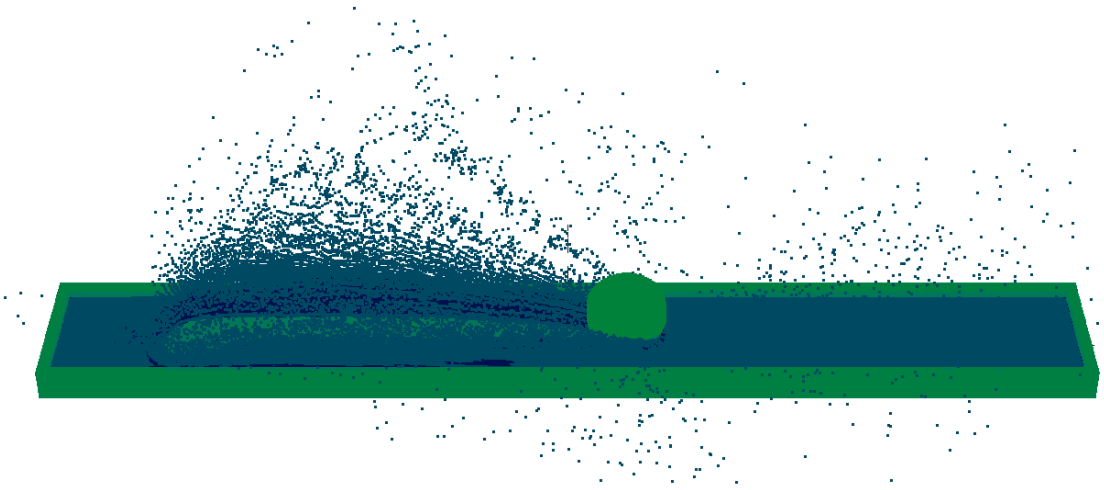
b): top view of the gear model

Figure 10.6: SPH model of the gear

The overall length of the runway is 8m, its width is 0.5m, and standing water height is 0.05m. Particle spacing of the inner fluid particles and boundary particles are the same which is 0.01m. The total number of particles is 442662. No-slip boundary condition is applied using the ISPH method with repulsive boundary force. The motion of each particle on the rotating wheel is a combination of a rotation about the mass centre and a forward translation. The theory of solid motion introduced in Chapter 3 is adopted. Different rolling velocities are considered. The pattern of the water spray with a rolling speed of 12.192m/s at different time instants is shown below in Figures 10.7 to 10.10.

Figure 10.7: Initial state of the landing gear





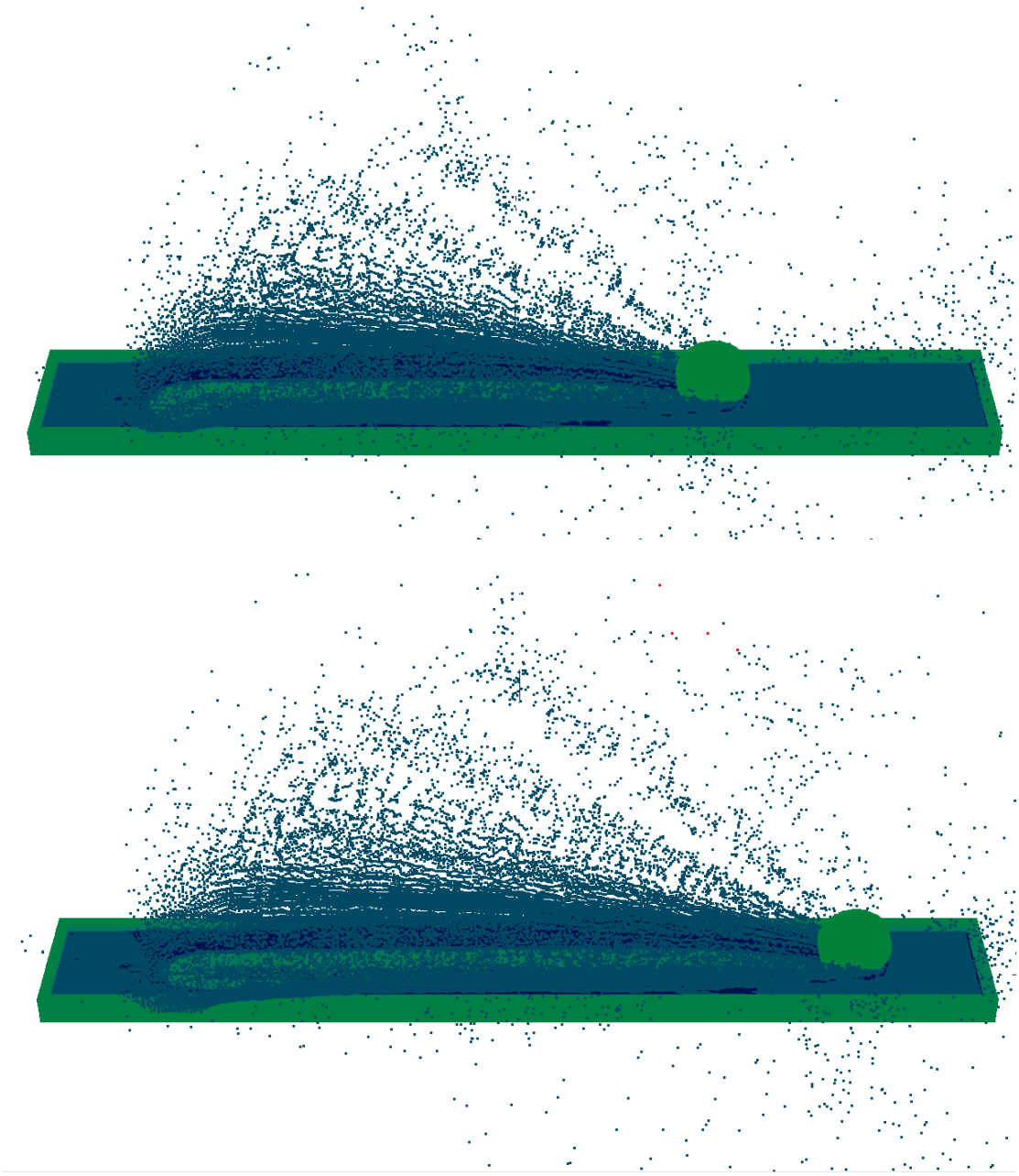


Figure 10.8: Pattern of water spray at time 0.05s, 0.1s, 0.15s, 0.2s and 0.25s

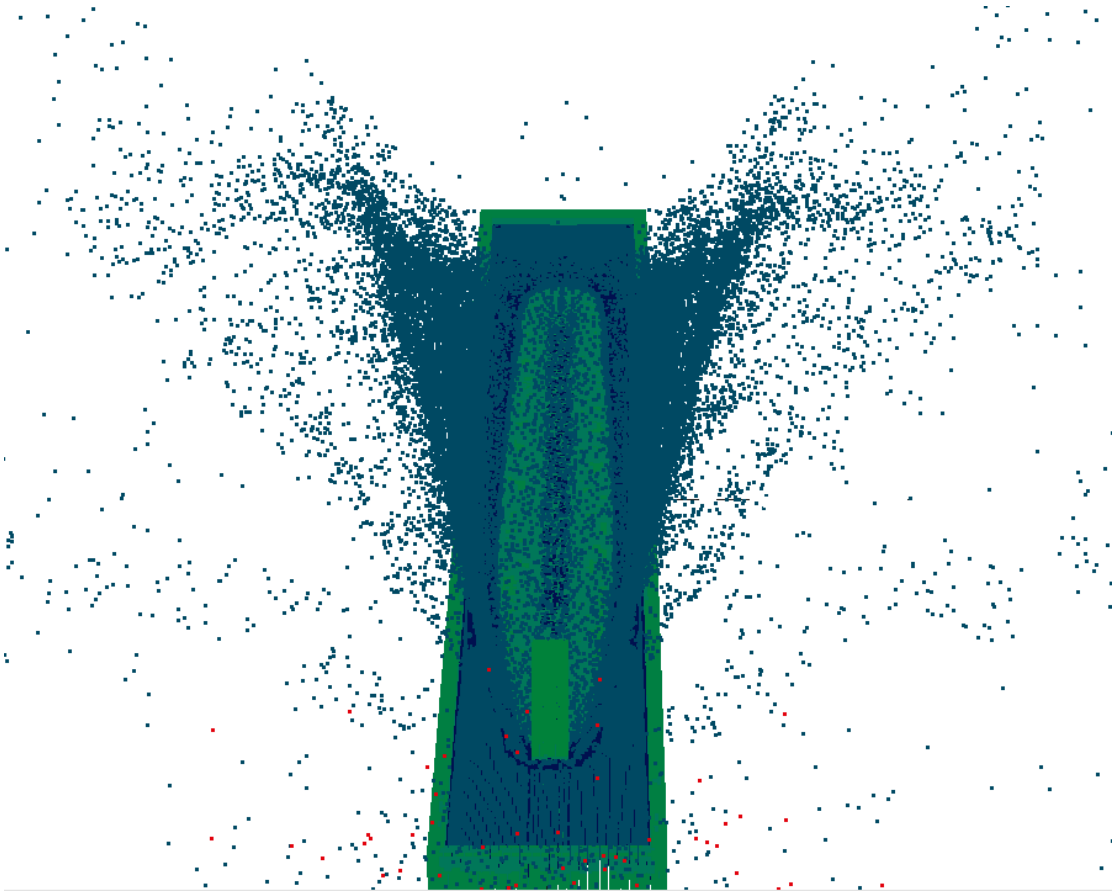


Figure 10.9: Top view of the water spray pattern at 0.25s

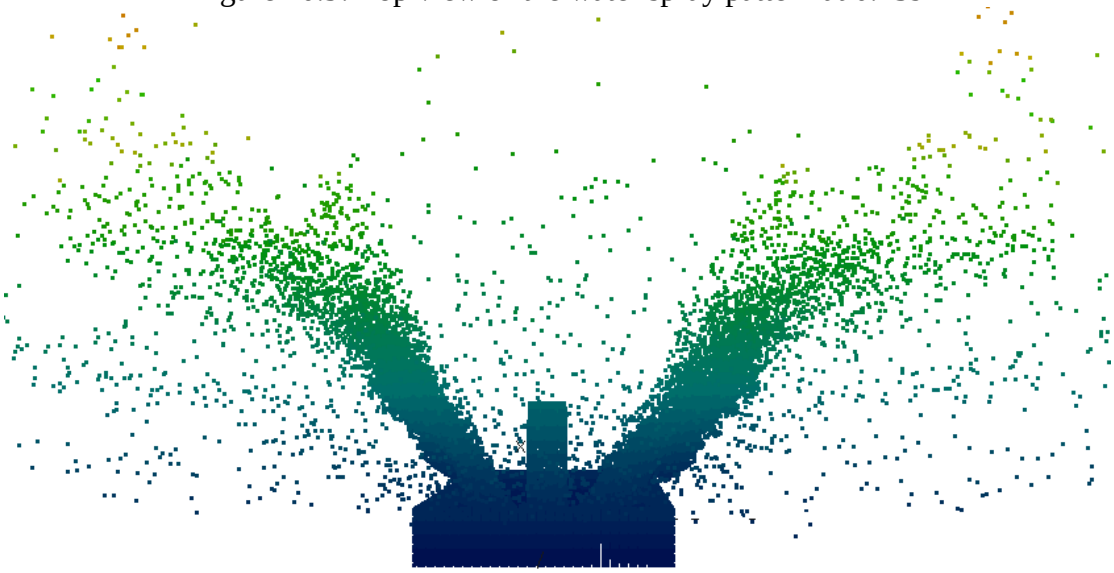


Figure 10.10: Front view of the water spray pattern at 0.25s

The pattern of water spray shown in the figures above is consistent with the description of water trajectory given in Daugherty & Stubbs (1987): '*The water in the path of the tyre footprint is almost completely displaced. Some of the water is expelled*

forward out of the footprint (bow wave) with low density. The major contributor to the volume of water which might be ingested by the engine is the water ejected laterally from the tyre footprint. As water is expelled laterally from the tyre footprint, it encounters an adjacent wall of water next to the tyre footprint edge, which absorbs some of the lateral energy. The collision causes the original laterally moving unit of water to change direction and to be thrown upwards. The next unit of water on the surface, having absorbed lateral impacting energy, undergoes the same process and is thrown upwards but with less initial velocity. Such action induced by the tyre produces a sheet of spray, as opposed to a circular jet, and the wake from the tyre on the surface, much like that from a boat, has enough lateral energy to propel a much larger amount of surface water into the air than in the direct path of the tyre footprint.'

It is the benefit of the meshless nature of the SPH method that the water spray can be simulated realistically. With the boundary treatment proposed for the ISPH method the fluid structure interaction problems can be simulated efficiently.

The maximum spray height from different rolling velocity is presented in table 10.1 and the related values of lateral and vertical components are shown in Figure 10.11.

Tire speed	Maximum spray height	CPU time
6.096m/s	0.608m	56484s
12.192m/s	1.168m	52968s
18.288m/s	1.9004m	51952s
24.384m/s	2.6582m	50369s

Table 10.1: spray height and CPU time for different rolling velocities

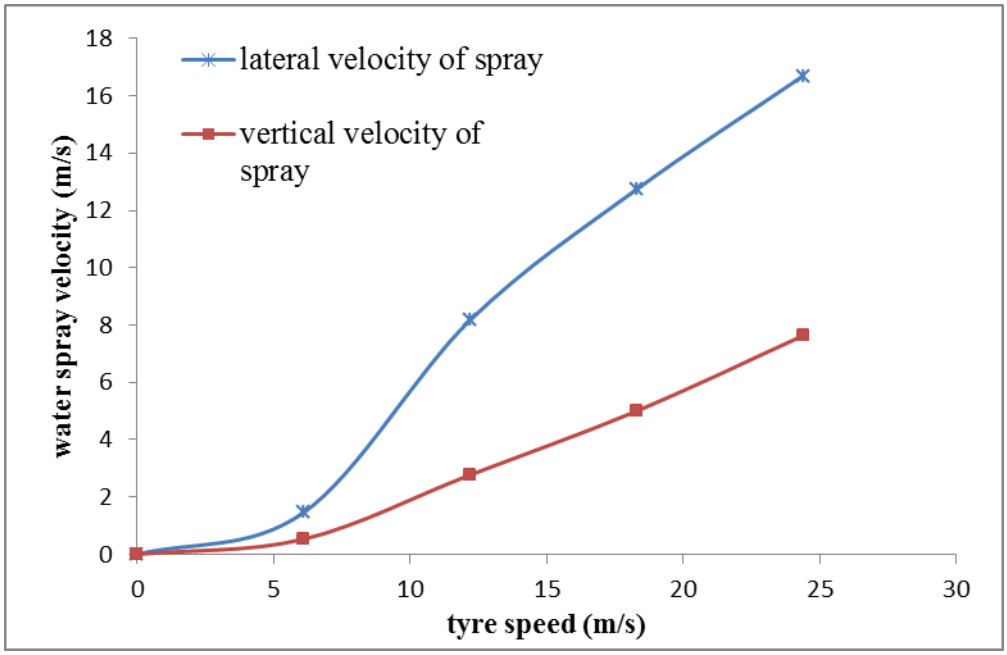
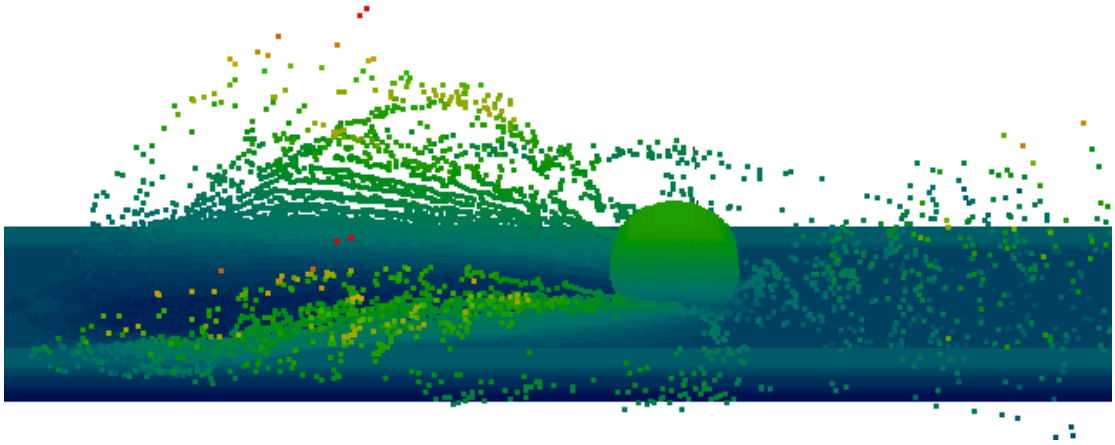


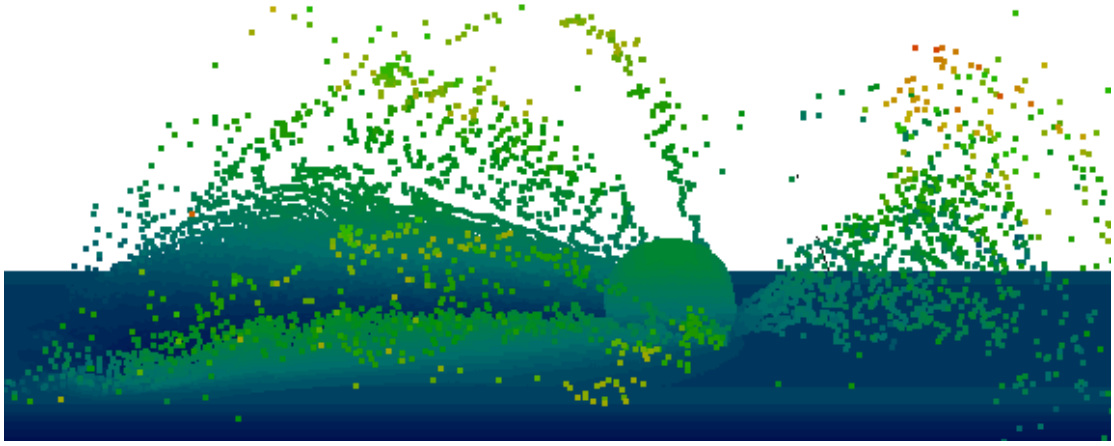
Figure 10.11: Spray rate caused by different tyre speeds

Naturally, higher rolling speed will produce higher water spray and therefore it is more likely that the water will be ingested into the engine. From the simulation point of view, higher velocity requires less computational time.

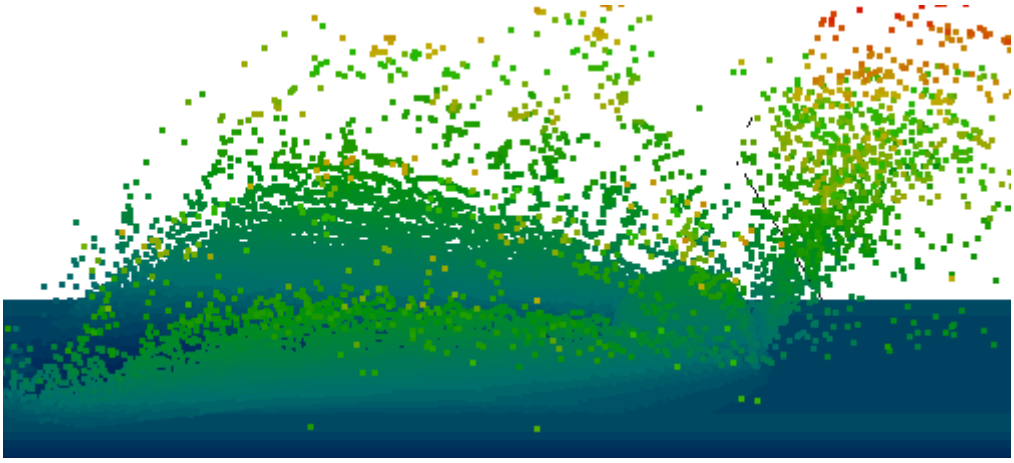
The influence of water depth is also considered. The spray patterns obtained based on three different water depths of 0.05m, 0.08m and 0.1m with the same rolling speed of 6.096m/s are depicted in Figure 10.12.



(a). water depth is 0.05m



(b). water depth is 0.08m



(c). water depth is 0.1m

Figure 10.12: water spray at t=0.25s with different water depths

The most noticeable difference that can be seen in Figure 10.12 is the height and volume of the “bow wave” which is the amount of water expelled forward out of the footprint.

Comparing Figure (a), (b) and (c) in Figure 10.12, it is clear that a shallow water depth will produce a weak bow wave. As the water depth increases the bow wave becomes stronger. The bottom of the fuselage will be impacted and the windscreen will be affected by the large amount of water spray. This will cause difficulties for the operation of the aircraft. The amount of the lateral spray rises with the increase of water depth as well. Hence, it is important to reduce the water depth on the runway as far as possible.

The maximum spray height as well as the CPU time required due to different water depth is listed in table 10.2

Water depth	Maximum spray height	Number of particles	CPU time
0.05m	0.608m	241656	25228s
0.08m	0.897m	327840	45642s
0.1m	0.956m	384642	60965s

Table 10.2: spray height and CPU time for different water depths

The CPU time required increases with the water depth and this is reasonable since more particles are used in the computation for a greater water depth. According to the values of the maximum spray height, the spray becomes stronger as the water depth increases. For a specific case in practice, the water spray pattern can be predicted using the numerical method developed so its impact to the operation and safety can be assessed.

10.4 Aircraft ditching simulation

Ditching is an emergency landing of an aircraft on water. The ability of an aircraft to remain afloat is important especially for rescue operations. In addition, the high impact loads created when the aircraft comes into contact with water may lead to damage of the structure and together with the violent decelerations this presents a substantial risk of severe injuries for passengers (Streckwall, et al. 2007). Thus the loads and motions of the aircraft during ditching need to be determined. The conventional approach to investigate aircraft ditching is either to carry out experiments in model scale or to adopt numerical approaches that deliver equivalent data.

Since SPH has advantage in simulating violent free surface flows, it should be suited for aircraft ditching simulation. And on the other hand, it is noted that the ditching process is a forward speed dominated flow problem involving structure moving on the surface of water. So it is appropriate to use this as an example to test the performance of SPH method for this type of problem.

The numerical model is based on the model aircraft tested by McBride & Fisher (1953) as shown in Figure 10.13. The test model is designed with a high-wing in order to eliminate the influence from the wing. The full model is 1.22m long, has a maximum radius of 0.1m for the cross section of the fuselage, the main wing spans 1.68m and the weight is 5.67kg. The mass centre is located at a distance of 0.53m away from the nose and the moments of inertia in the principle axes are also given in the report of the test, which are

$I_x = 0.293352 kgm^2$, $I_x = 0.293352 kgm^2$, $I_x = 0.527952 kgm^2$.

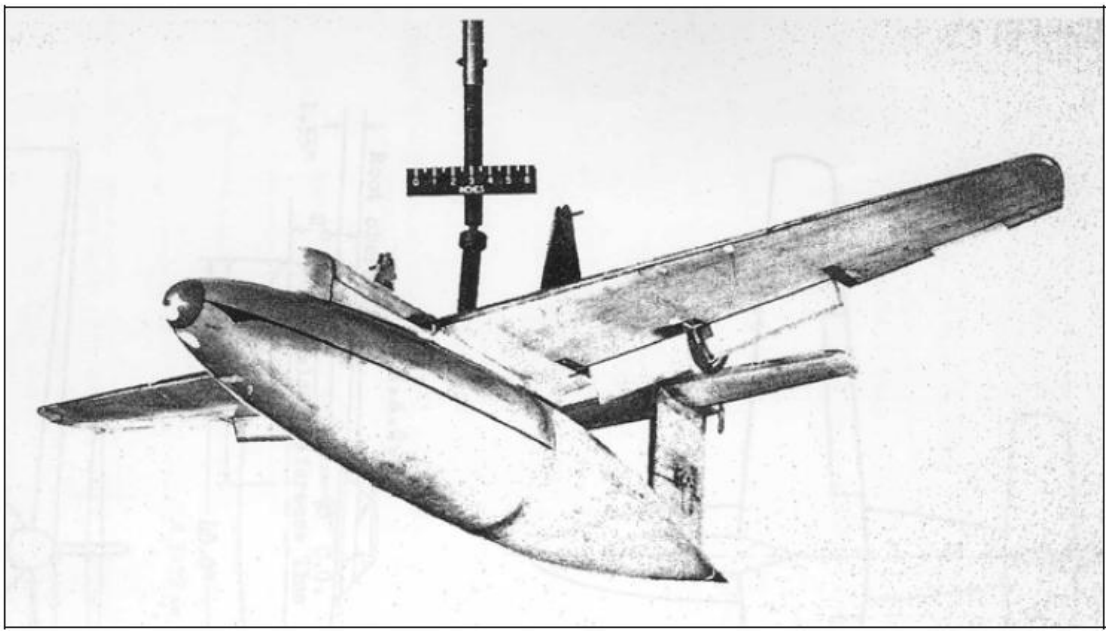


Figure 10.13: Original model configuration (McBride & Fisher April 1953)

In the simulation, the water block is 16m long, 2m wide and 0.5m deep. The aircraft and water are both represented using particles with a same particle spacing of 0.05m. The time step size is 10^{-4} s. No-slip boundary condition is applied. On the fixed wall, repulsive force and wall pressure are both used whereas on the aircraft only the pressure of the particles is considered.

In order to test the capability of the developed algorithm for ditching simulation, a simplified situation of a floating aircraft having zero speed, with an attitude of 0^o and 10^o, as shown in Figures 10.14 and 10.15 are considered first. The basic numerical

model setting is the same as that in the ditching test, only the length of the water block is reduced to 8m to reduce the particles involved in the computation.

Figure 10.14: floating aircraft with an attitude of 0^o

Figure 10.15: floating aircraft with an attitude of 10^o

In theory, the aircraft floating on the surface of water will start to sink because of the gravity force. After it goes into water it will experience pressure force from the water. Thus, the velocity of the aircraft will change as a result of these forces. Besides, since the geometry of the aircraft is not symmetric in both *x* and *z* directions, there will be a resultant moment from the pressure about the mass centre, which will cause the aircraft to rotate. The calculated velocities and attitude during the floating process are shown in Figures 10.15 and 10.16.

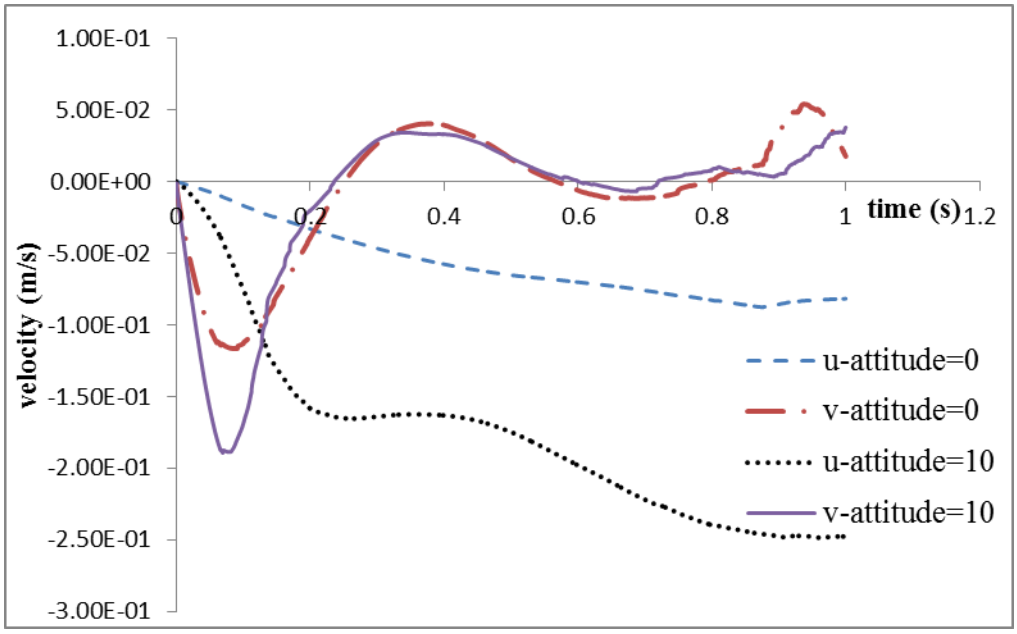


Figure 10.16: Velocity of the aircraft with different initial attitude

From Figure 10.16 we can see that the downward velocity increases in the beginning because of the gravity force. After some time the aircraft sinks deeper into the water and the downward velocity decreases because of the increased buoyancy. And then the overall vertical velocity fluctuates slightly around zero. This means the gravity force and buoyancy balances each other. The surge velocity is negative and the value increases from the beginning to 0.9s for both two cases. The acceleration is larger and fluctuates more frequently when the initial attitude is 10^{o}.

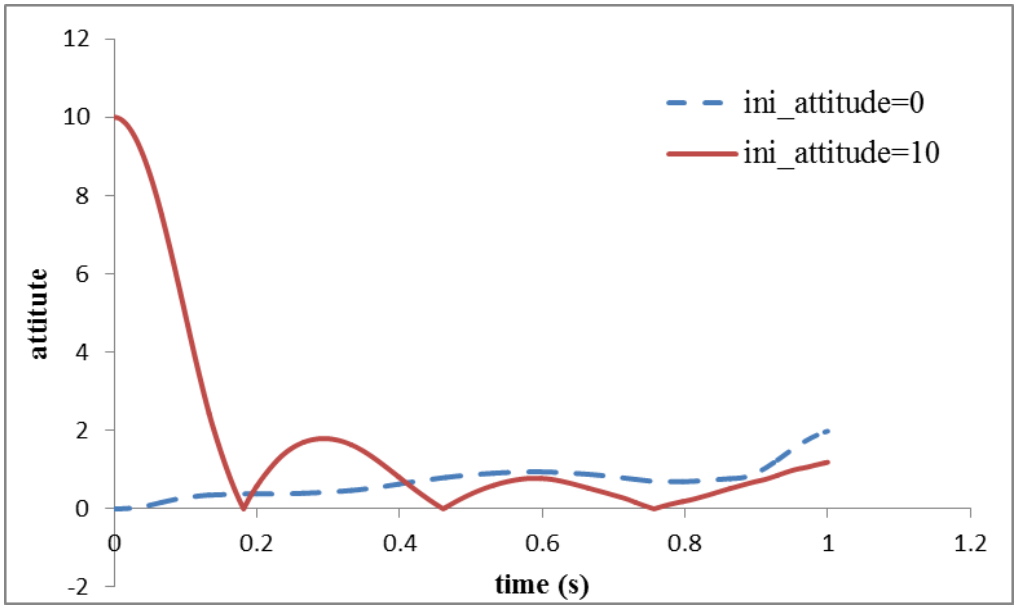
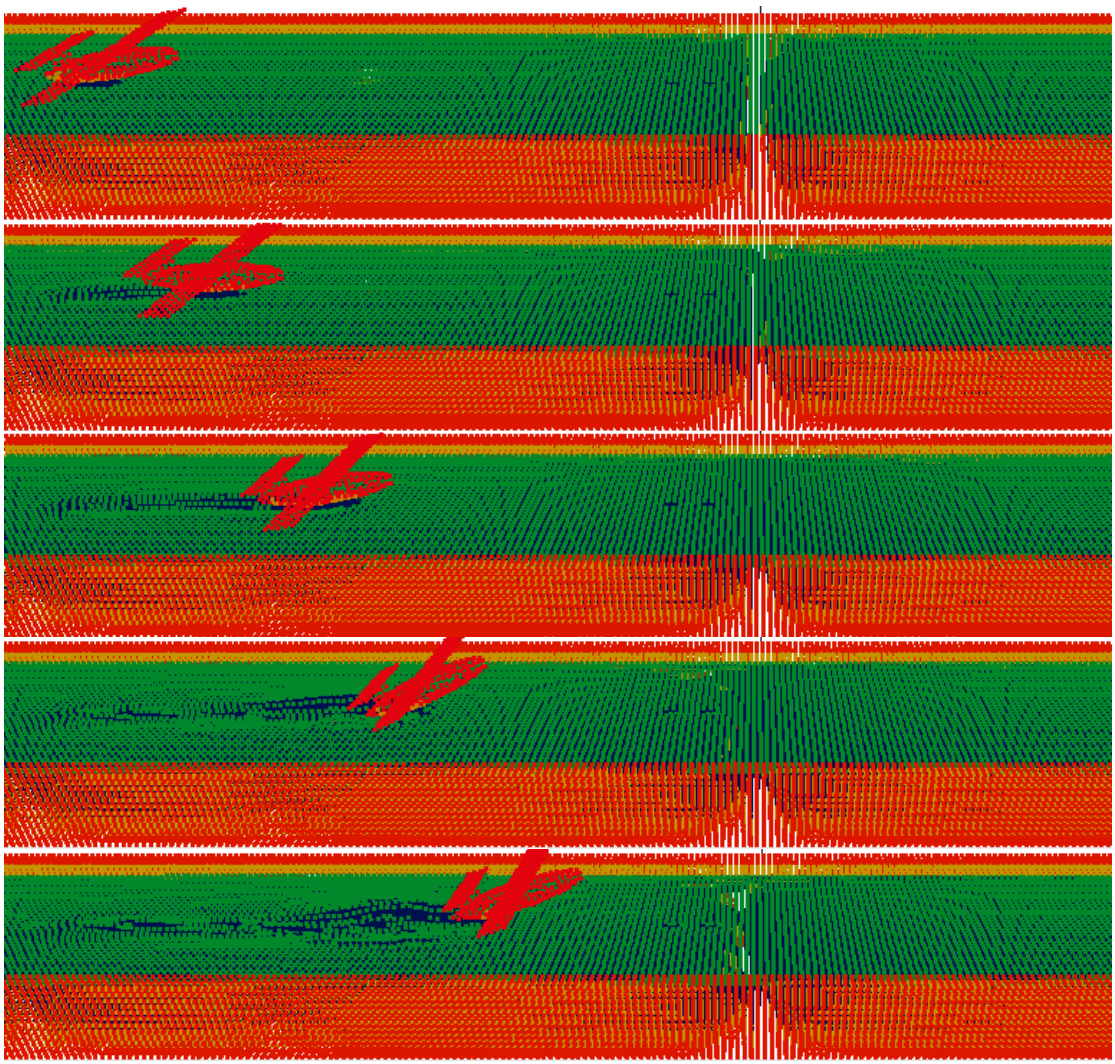


Figure 10.17: Attitude of the aircraft with different initial attitude

It shows in Figure 10.17 that for the case with an initial attitude of 10^o the attitude of the aircraft decreases rapidly in the beginning. Afterwards the value of the attitude fluctuates slightly around zero just the same as the vertical velocity. For the case with an initial attitude of 0^o, the value of the attitude does not change significantly.

In brief, the values of velocity and attitude look reasonable so the simulation of the floating aircraft is regarded to be successful.

Next, a ditching process with s a landing speed of 9.14m/s and an initial attitude of 10^o is simulated using the algorithm. The total number of particles is 190722, total CPU time is 43555s for 1s simulation. The snapshots of the aircraft and the flow pattern during ditching are shown in Figure 10.18.



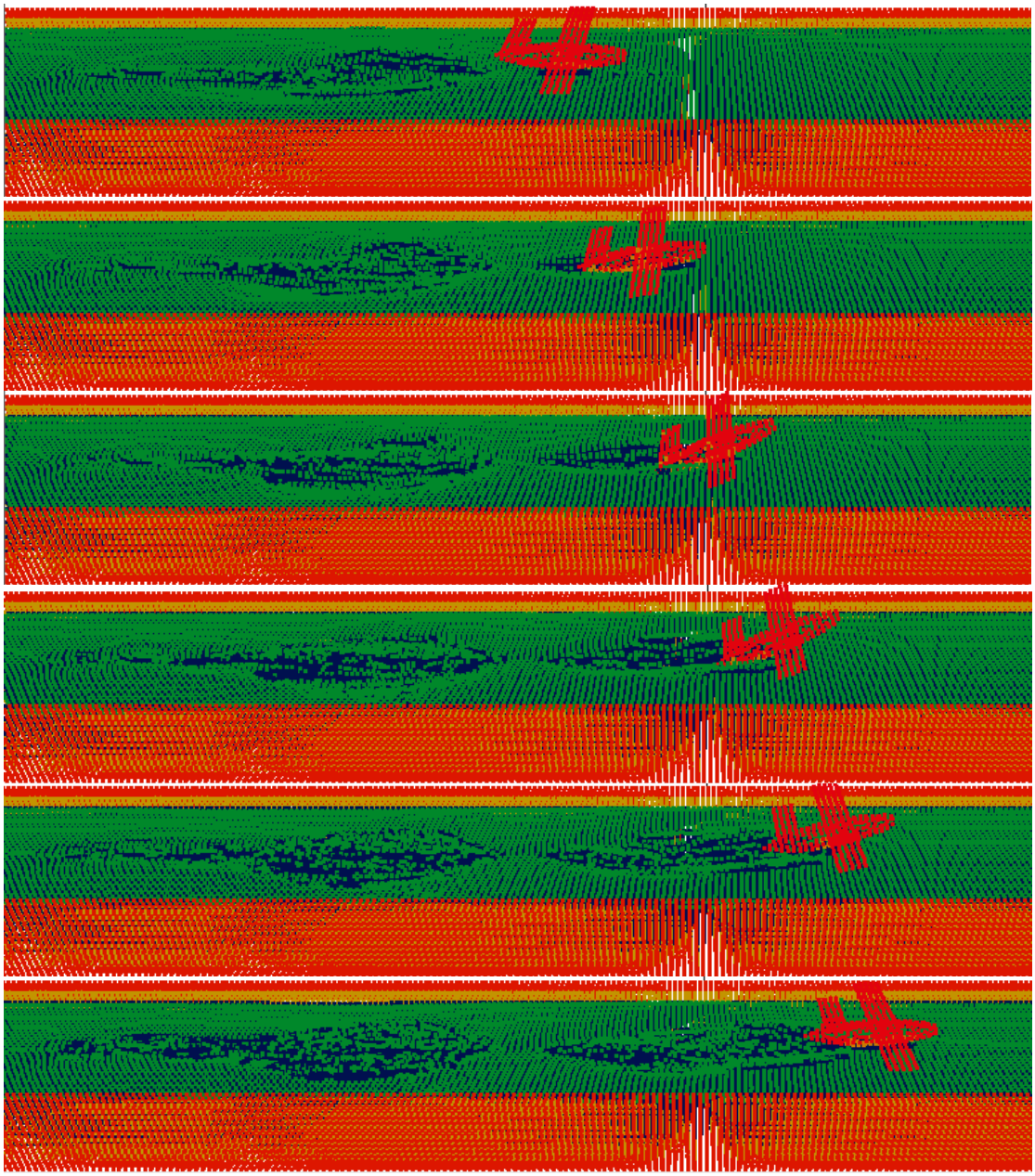


Figure 10.18: snapshots of ditching at time t=0, t=0.1s, t=0.2s, t=0.3s, t=0.4s, t=0.5s, t=0.6s, t=0.7s, t=0.8s, t=0.9s, t=1s

From Figure 10.18 it seemd that the aircraft bounces up and down after it made contact with the water during the process. Typically, in the period between 0.5s and 0.6s, the aircraft is out of water flying in the air. The value of velocity and attitute of the aircraft is recorded and dispalyed in curves shown in Figures 10.19 and 10.20.

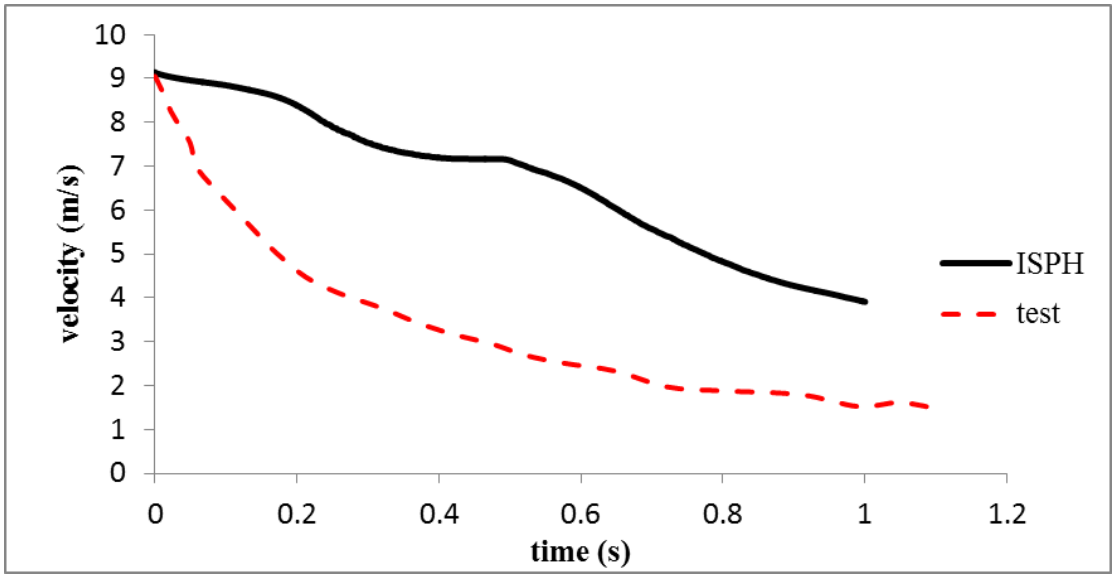


Figure 10.19: Forward velocity during ditching

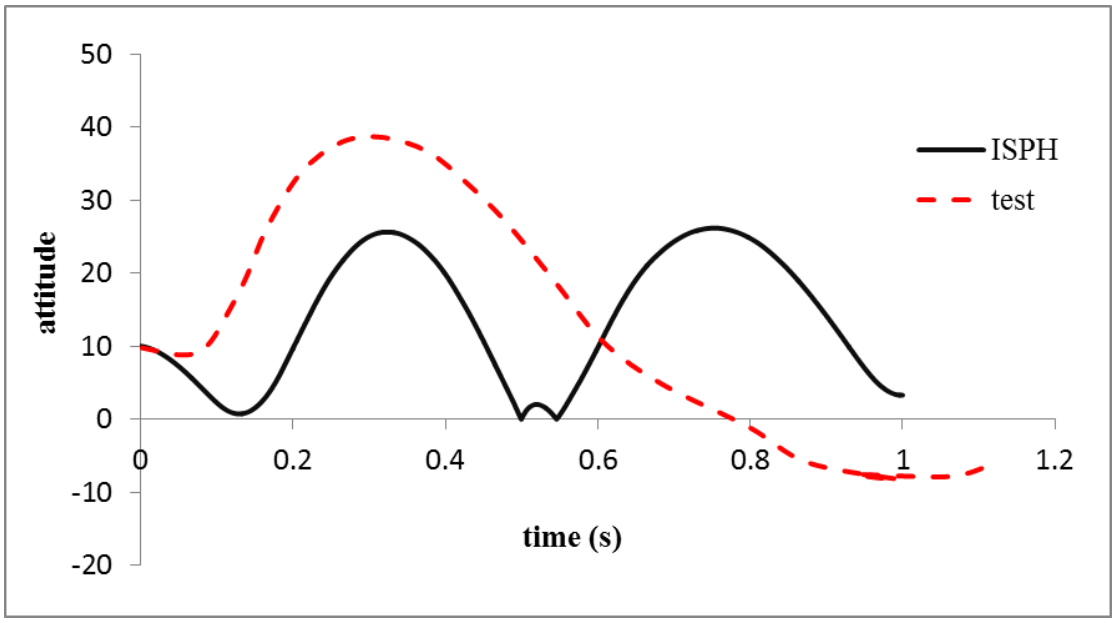
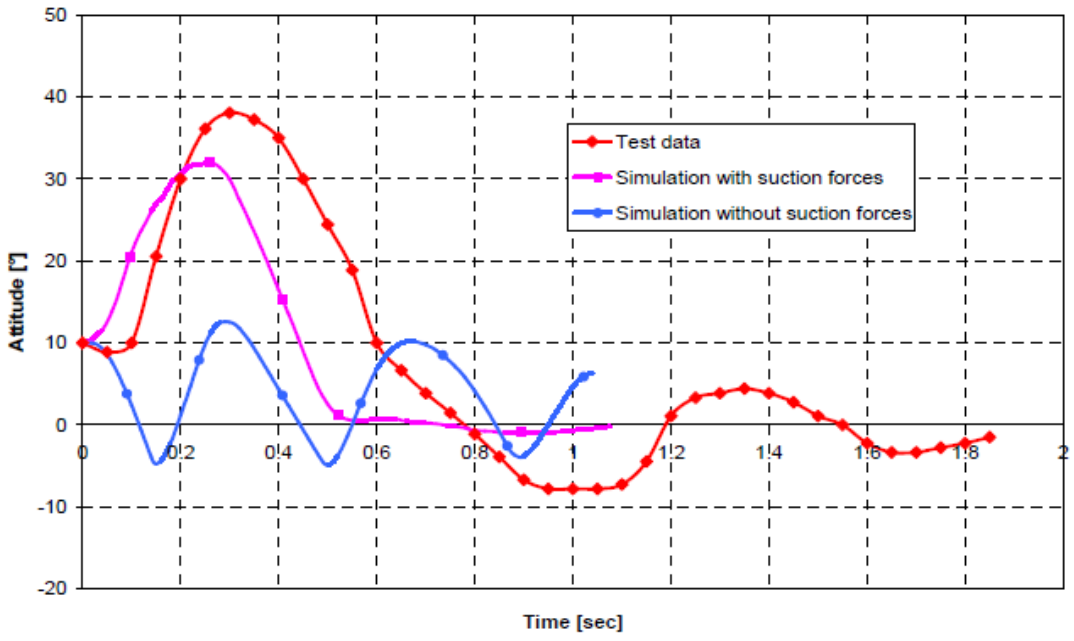


Figure 10.20: Attitude during ditching

Figure 10.19 shows the variation of the velocity during both ditching simulations the test whereas Figure 10.20 depicts the attitude of the aircraft model. It can be observed that during the experiment, the aircraft attitude increases strongly from 0.1s till 0.35s and then decreases to minus 10^{o} at 1s. Correspondingly, the velocity decreases rapidly from the beginning till 0.35s, and then decreases less rapidly afterwards. By contrast, the attitude of the aircraft obtained from the ISPH simulation fluctuates more. It decreases rapidly in the first 0.1s and then increases to 28^{o} at 0.35s.

Afterwards it decreases till 0.5s and another increase is observed. It seems that the aircraft flies up into air at that moment. The attitude increases and decreases in the next period of 0.4s. The corresponding velocity decreases more slowly in the simulation than in the experiment.

The difference between the results from the experiment and ISPH simulation can be attributed to the problem that no suction force is modelled by the current ISPH algorithm (Climent, et al. 2006; Streckwall, et al. 2007; Toso 2009; Zhang, et al. 2012). In reality, the velocity of the water flow increases around the immersed part of the fuselage, which causes the pressure to decrease according to Bernoulli's equation. Although same principle applies to air flow too, the presure on the wetted part of the fuselage should decrease more as water has a larger density and so a suction force is generated. The results obtained by Toso (2009) considering the effect of the suction force is displayed in Figure 10.21 as a reference.



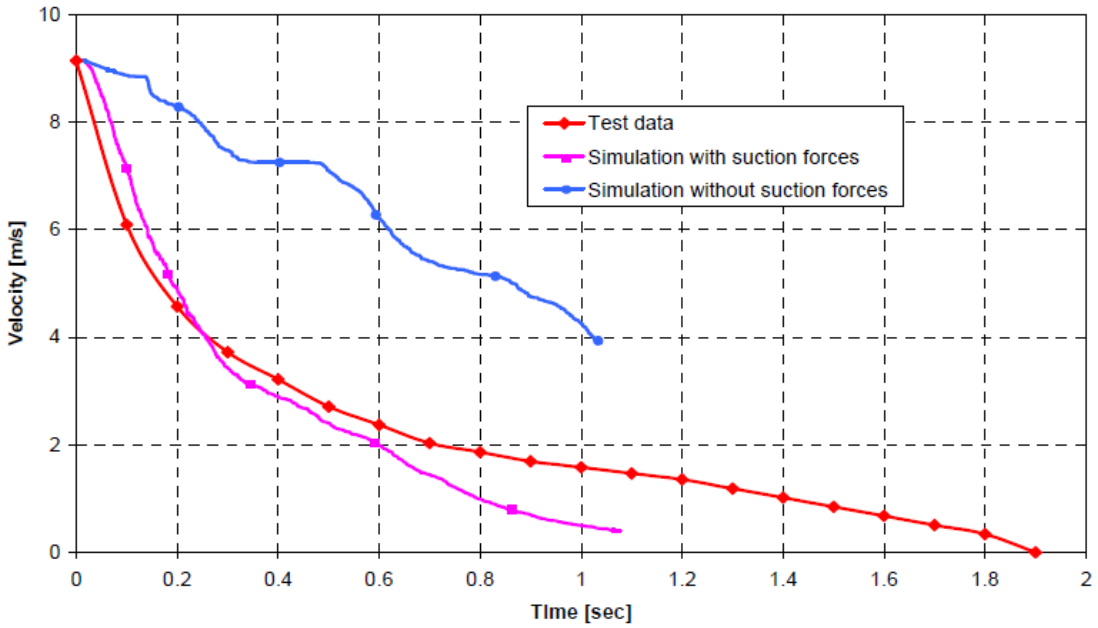


Figure 10.21: Reference results obtained by Toso (2009)

From the results shown in Figure 10.21, it is clear that suction force is a crutial factor for an correct simulation. It was reported that SPH method could not produce negative pressure hence it failed in the prediction of suction force (Climent, et al. 2006); more research is still required to eliminate the deficiencies of SPH method for modelling of fluid solid coupling motions dominated by forward speed.

Chapter 11 Conclusion and Future work

SPH is a promising tool for FSI simulations because of the meshless nature of the method. The understanding of the concept is easy and it can be efficiently applied to incompressible fluid flows. This project focuses on the capability of the SPH method for simulating incompressible fluid flows interacting with a rigid body.

The fundamental concept of SPH is reviewed including the approximation formulizations and implementations. The neighbouring particle search algorithm is very important for the computational efficiency of SPH method since it is the most time consuming operation in a SPH simulation. The cell-link Verlet list method is selected. Unlike the conventional method which considers only half of the adjacent cells, the pairing characteristic of the particles is considered so only half of the particles inside the adjacent cells are searched and this is more efficient.

First of all, the effect of the key factors of SPH method such as the choice of kernel functions, smoothing length of the kernel function, particle numbers and time step size is studied in detail with one dimensional cases. It is found that increasing particle numbers normally leads to better results but different kernel functions and different time step sizes do not significantly affect the result. SPH has a second order accuracy with respect to smoothing length and reducing smoothing length can improve the accuracy but the smoothing length needs to be given in a certain range in relation to the particle spacing to ensure that there are enough neighbouring particles.

The performances of weakly compressible SPH method and incompressible SPH method with explicit solution approach and implicit solution approach are compared according to flow patterns, pressure distributions and CPU time consumptions in the dam breaking case studies and multi-phase flow examples. All these methods can provide reasonable flow patterns of the specific motions but the distribution of particles in WCSPH is normally less uniform than ISPH methods. Pressure distribution obtained from the WCSPH method is quite erratic even for water in hydrostatic equilibrium whereas correct pressure distributions can be produced by

using ISPH with either the explicit or implicit solution approach. Using the explicit solution approach for ISPH method requires the least CPU time in most cases so it is considered to have great potential for the future applications.

Two boundary treatments, namely to use repulsive force or to use denser wall particles, are proposed for the incompressible SPH method. Although these two boundary treatments have been used in WCSPH, they have not been used for ISPH in the known literature. The accuracy of the prediction of pressure obtained from ISPH method using these two boundary treatments are investigated in a dam breaking flow simulation. Subsequently, the performance of these two boundary treatments with ISPH method is demonstrated with two 2D examples including wedge dropping and spring supported dam. The results obtained agree with experimental data. These two treatments can be used efficiently for more complicated engineering problems related to fluid structure interaction problems as the model can be generated just the same as the geometry without additional ghost particles. The difficulty associated with model generation is thus largely reduced and fewer particles need to be used compared to ghost particles treatment especially in 3 dimensional problems.

A new method combining WCSPH and ISPH for multi-phase flow is developed to study the effect of air on the flow. Dam breaking case is studied under this situation but no significant influence is found from the air. Rising air bubble in water and Rayleigh-Taylor instability problem are simulated to test the performance of this new combined method. It is clear that this new method is able to provide a good prediction of the fluid motion. The initial idea of this method is to retain the compressible nature of each fluid and to save computation time at the same time. However, it requires more computational time than any of the individual methods, when the ISPH method with implicit solution approach is combined with the WCSPH method. Hence, this approach is not recommended. On the other hand, the combined method of WCSPH with ISPH method using explicit solution approach produces good results and requires less computational time.

A 3D computer code is developed for a more realistic simulation of general fluid rigid body interactions. Dam breaking and wedge dropping cases are simulated to test the performance of the algorithm. The spray caused by aircraft landing gear running

on a wet runway, a typical case which is difficult for the traditional mesh required numerical method, is studied using SPH. Different standing water depth and rolling speed are considered. It is found that with greater water depth or higher rolling speed, the amount of spray generated is also greater both in lateral and vertical directions. Aircraft ditching is simulated using this 3D algorithm as well. However, the forward speed dominated motion remains a challenge to SPH method because it is difficult to capture the correct suction force.

Contributions:

- The performance of two simple boundary treatments, i.e. repulsive force and denser wall particles, proposed for the ISPH method is investigated and the efficiency confirmed.
- The explicit solution approach is investigated for the ISPH method.
- A new method is proposed for air water two-phase flow by combining WCSPH for compressible fluid and ISPH for incompressible fluid.
- A 3D computer code has been developed for general fluid rigid body interaction problems using SPH for the entire system.

Future work

In order to improve the overall performance of the SPH method for different engineering problems, more in-depth analysis and assessment should be conducted in the following areas.

Energy dissipation and momentum conservation of the methods need to be analysed systematically.

Stability and consistency of the algorithm using existing boundary treatments should be tested thoroughly.

Higher order time stepping scheme and correction terms such as the ones used in XSPH (Monaghan 1989; 1992; 2002) can be considered to improve the newly developed method in order to produce more accurate results for simulation of multi-phase flows.

Different values rather than zero can be adopted as the reference pressure in the calculation and an investigation of such modification to the prediction of suction force can be conducted.

For complex 3D problems, small particle spacing and large number of particles are required, which demands high computational resources. Therefore, the current program needs to be converted into a parallel code and computer cluster can be used in the simulation.

To expand the applicability of the current program to a wider range of hydrodynamic problems, implementation of inlet and outlet boundary conditions should be considered.

Models for deformable solid and coupling strategy should be developed to extend the application of these algorithms.

Appendices

Formulation of SPH

The basic formulation of SPH will be discussed in this section. As mentioned before, SPH is based on the theory of integral interpolant (Monaghan 1987; 1988; 1989; Monaghan & Kocharyan 1995; Liu & Liu 2003b), a general function $A(\mathbf{x})$ can be reproduced as

$$A(\mathbf{x}) = \int_{\Omega} A(\mathbf{x}')\delta(\mathbf{x}-\mathbf{x}')d\mathbf{x}' \tag{A.1}$$

where Ω is the volume of the integration, and $\delta(\mathbf{x}-\mathbf{x}')$ is the Dirac delta function defined as

$$\delta(\mathbf{x}-\mathbf{x}') = \begin{cases} \infty.....\mathbf{x}=\mathbf{x}' \\ 0.....\mathbf{x}\neq\mathbf{x}' \end{cases} \tag{A.2}$$

and it also satisfies the following unity condition

$$\int_{-\infty}^{\infty} \delta(\mathbf{x}-\mathbf{x}')d\mathbf{x} = 1 \tag{A.3}$$

Equation (A.1) is exact but not practically useful. The concept of SPH is to replace Dirac delta function with a kernel function to obtain an approximation (Monaghan 1982; Monaghan & Gingold 1983; Monaghan & Poinracic 1985a; Liu & Liu 2003b) and the kernel estimation denoted by $\langle A(\mathbf{x})\rangle$, is defined as

$$\langle A(\mathbf{x})\rangle = \int_{\Omega} A(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{A.4}$$

where $W(\mathbf{x}-\mathbf{x}')$ is the kernel function and h is the smoothing length which defines the influence domain of the kernel function. Similar process can be applied to the gradient of function approximation

$$\langle \nabla\cdot A(\mathbf{x})\rangle = \int_{\Omega} (\nabla\cdot A(\mathbf{x}'))W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{A.5}$$

Via integration by part, this can be expressed as

$$\langle \nabla \cdot A(\mathbf{x}) \rangle = \int_{\Omega} \nabla \cdot \left(A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) \right) d\mathbf{x}' - \int_{\Omega} A(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \qquad \text{(A.6)}$$

Applying divergence theorem (Batchelor 1973) to the first integral in Equation (A.6)

$$\langle \nabla \cdot A(\mathbf{x}) \rangle = \int_{s} A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) \cdot \mathbf{n} d\mathbf{x}' - \int_{\Omega} A(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \qquad \text{(A.7)}$$

where $\mathbf{n}$ is the unit vector normal to the surface, Ω is the volume of the integral. The first item on the right hand side is normally zero due to the compact property of kernel function which will be discussed in section 2.4. Therefore, Equation (A.5) becomes

$$\langle \nabla \cdot A(\mathbf{x}) \rangle = -\int_{\Omega} A(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \qquad \text{(A.8)}$$

On the left hand side of the Equation (A.8) the derivative is taken with respect to $\mathbf{x}$ while one the right hand side it is taken with respect to $\mathbf{x}'$. From the above equation, it can be seen that the differential operation on a function is transferred to a differential operation on the smoothing function. And the SPH integration allows the spatial derivative of a function to be calculated based on the values of the function and the derivative of the kernel function which can be calculated analytically, instead of the function derivative itself. This reduces the consistency requirement and produces more stable solutions for PDE (Liu 2002).

In order to facilitate numerical approximation, the infinitesimal volume $d\mathbf{x}'$ in the integral Equation (A.8) is replaced by the particle volume which can be expressed using mass m and density ρ,

$$dx' = \frac{m_b}{\rho_b} \qquad \text{(A.9)}$$

The SPH particle approximation form can be derived if the integration is approximated by a summation over the neighbouring particles which are located within the smoothing length domain

$$\langle A(\mathbf{x})\rangle_{\mathrm{a}} = \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \frac{A_{\mathrm{b}}}{\rho_{\mathrm{b}}} W(|\mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\mathrm{b}}|, h) \tag{A.10}$$

The subscript a indicates the specific particle and b indicates neighbouring particles and N is the total number of particles inside the smoothing domain. Similarly, the approximation for spatial derivatives is obtained by representing the integration in Equation (A.8) with the sum of the contribution from discrete particles

$$\langle \nabla A(\mathbf{x})\rangle_{\mathrm{a}} = \sum_{\mathrm{b}=1}^{N} m_{\mathrm{b}} \frac{A_{\mathrm{b}}}{\rho_{\mathrm{b}}} \nabla W(\mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\mathrm{b}}, h) \tag{A.11}$$

The minus sign in Equation (A.8) is removed because the derivative is in terms of x_{b} instead of x_{a}. This equation implies that the derivatives of any function can be found by differentiating the kernel rather than by using grids. As a consequence, instead of solving partial differential equations for hydrodynamics problems, only ordinary differential equations need to be solved

References

Abdolmaleki, K., K.P. Thiagarajan & M.T. Morris-Thomas (2004) 'Simulation of the dam break problem and impact flows using a Navier-Stokes solver', in, *15th Australasian Fluid Mechanics Conference* (Sydney, Australia: The University of Sydney).

Adami, S., X.Y. Hu & N.A. Adams (2010), 'A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation', Journal of Computational Physics 229:5011-5021.

Allen, M.P. & D.J. Tildesley (1990), Computer simulation of liquids, (New York: Oxford University Press).

Aluru, N.R. (1999), 'A reproducing kernel particle method for meshless analysis of microelectromechanical systems', Computational Mechanics 23:324-338.

Antoci, Carla, Mario Gallati & Stefano Sibilla (2007), 'Numerical simulation of fluid-structure interaction by SPH', Comput. Struct. 85(11-14):879-890.

Attawy, S.W., M.W. Heinstein & J.W. Swegle (1994), 'Coupling of smoothed particle hydrodynamics with the finite element method', Nuclear Engineering and Design 150:199-205.

Awile, Omar., Ferit. Büyükkeçeci, Sylvain. Rebouxa & Ivo F. Sbalzarini (2012), 'Fast neighbor lists for adaptive-resolution particle simulations', Computer Physics Communications 183:1073-1081.

Bairstow, L. & A. Fage (1916) 'Oscillations of the tailplane and body of an aeroplane in flight', in, *ARCR. and M* (In: technical Report of the Advisory Committee for Aerodynamics for the year 1916-1917. II, His Majesty's Stationary Office, Lodon: 461-467.

Basa, Mihai, Nathan J. Quinlan & Martin Lastiwka (2009), 'Robustness and accuracy of SPH formulations for viscous flow', International Journal for Numerical Methods in Fluids 60(10):1127-1148.

Batchelor, G.K. (1973), An Introduction to Fluid Dynamics (Cambridge Cambridge University press).

Bathe, K.-J. & H. Zhang (2004), 'Finite element developements for general fluid flows with structural interactions', International Journal for Numerical Methods in Engineering 43:785-819.

Bathe, K., C. Nitikitpaiboon & X. Wang (1995), 'A mixed dispalcement-based finite element formulation for acoustic fluid structure interaction', Comput. Struct. 56:225-237.

Batra, R.C. & G.M. Zhang (2008), 'Modified smoothed particle hydrodynamics basis functions for meshless methods, and their application to axisymmetric Taylor impact test', Journal of Computational Physics 227:1962-1981.

Belytschko, T., D.P. Flanagan & J.M. Kennedy (1982), 'Finite element methods with user-controlled meshes for fluid-structure interaction', Comput. Meth. Appl. Mech. Engng 33:669-688.

Belytschko, T., Y. Krongauz, D. Organ, M. Fleming & P. Krysl (1996), 'Meshless methods: An overview and recent developments', Computer Methods in Applied Mechanics and Engineering 139(1-4):3-47.

Belytschko, T., W.K. Liu & B. Moran (2000), Nonlinear finite elements for continua and structures, (New York, USA: Wiley and Sons Ltd).

Bishop, R.E.D. & W.G. Price (1979), Hydroelasticity of ships: Cambridge University Press).

Bishop, R.E.D., W.G. Price & Y. Wu (1985) 'General linear hydroelasticity theory of floating structures moving in a seaway', in, *Philosophical Transactions of the Royal Society of London.* : 375-426.

Bonet, J., S. Kulasegaram, M. X. Rodriguez-Paz & M. Profit (2004), 'Variational formulation for the smooth particle hydrodynamics (SPH) simulation of fluid and solid problems', Computer Methods in Applied Mechanics and Engineering 193(12-14):1245-1256.

Bonet, J. & T. S. L. Lok (1999), 'Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations', Computer Methods in Applied Mechanics and Engineering 180(1-2):97-115.

Bui, Ha H., K.Sako & R. Fukagawa (2007), 'Numerical simulation of soil-water interaction using smoothed particle hydrodynamcis (SPH) method', Journal of Terramechanics 44:339-346.

Capuzzo-Dolcetta, Robert & Robert Di Lisio (2000), 'A criterion for the choice of the interpolation kernel in smoothed particle hydrodynamics', Appl. Numer. Math. 34(4):363-371.

Carlton, Adam (2004), 'A summary study on arbitrary Lagrangian-Eulerian: methodology, implementation, and application', 2004 Summer Research Experience for Undefraduates, Oregon State University.

Chen, J. K. & J. E. Beraun (2000), 'A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems', Computer Methods in Applied Mechanics and Engineering 190(1-2):225-239.

Chen, J. K., J. E. Beraun & T. C. Carney (1999a), 'A corrective smoothed particle method for boundary value problems in heat conduction', International Journal for Numerical Methods in Engineering 46(2):231-252.

Chopra, A.K (1970), 'Earthquake Response of Concrete Gravity Dams ', Journal of Engineering Mechanics Division 96:443-454.

Chui, C.K. (1992) 'An introduction to wavelets', in, *Academic* (New York.

Climent, H., L. Benitez, F. Rosich, F. Rueda & N. Pentecote (2006) 'AIRCRAFT DITCHING NUMERICAL SIMULATION', in, *25th Congress of International Council of the Aeronautical Sciences* (Hamburg, Germany

Colagrossi, Andrea, Matteo Antuono & David Le Touze (2009), 'Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics', PHYSICAL REVIRE E 79.

Colagrossi, Andrea & Maurizio Landrini (2003), 'Numerical simulation of interfacial flows by smoothed particle hydrodynamics', Journal of Computational Physics 191(2):448-475.

Crespo, A.J.C., M. Gomez-Gesteira & R.A. Dalrymple (2007), 'Boundary conditions generated by Dynamic paticles in SPH methods', CMC 5:173-184.

Cummins, Sharen J. & Murray Rudman (1999), 'An SPH projection method', J. Comput. Phys. 152(2):584-607.

Dalrymple, Robert A. & Omar Knio (2001) 'SPH Modelling of Water Waves', in Hanson Hans & Larson Magnus (eds): ASCE): 80.

Daugherty, Robert H. & Sandy M. Stubbs (1987) 'Meaurments of flow rate and trajectory of aircraft tire-generated water spray', in, *NASA technical paper 2718*.

Denis, St. & W.J. Pierson (1953), 'On the motion of ships in confused seas', Trans. SNAME. 63:1955.

Dominguez, J.M., A.J.C. Crespo, M. Gomez-Gesteria & J.C. Marongiu (2010), 'Neighbour lists in smoothed particle hydrodynamics', International Journal for Numerical Methods in Fluids 67(12):2026-2042.

Donea, J., P. Fasoli-Stella & S. Giuliani (1977) 'Lagrangian-Eulerian finite element techniques for transient fluid-structure interaction problems', in, *Trans. 4th SMIRT Conf* (San Francisco.

Du, Ying., Xijun. Liu & Qifen. Jia (2004), 'Research about liquid solid interaction dynamics problems', Machine Tool & Hydraulics 11:9-12.

Ellero, Marco, Mar Serrano, Pep Espanol, \#241 & Ol (2007), 'Incompressible smoothed particle hydrodynamics', J. Comput. Phys. 226(2):1731-1752.

Faltinsen, O.M. (1993), 'Sea loads on ships and offshore structures', Cambridge Ocean Technology Series:282.

Faltinsen, Odd M., Maurizio Landrini & Marilena Greco (2004), 'Slamming in marine applications', Engineering Mathematics 48:187-217.

Fossen, Thro I. (2002), Marine control systems: guidance, navigation, and control of ships, rigs and underwater vehicles: Marine Cybernetics AS.).

Fulk, D.A. & D.W. Quinn (1996), 'An analysis of 1-D smoothed particle hydrodynamics kernels', Journal of Computational Physics 126:165-180.

Gesteria, M.G., B.D. Rogers, R.A. Dalrymple, A.J.C. Crespo & M. Narayanaswamy (2010), 'User Guide for the SPHysics code'.

Gingold, R.A. & J.J. Monaghan (1977), 'Smoothed Particle Hydrodynamics : Theory and Application to Non-Spherical stars', Monthly Notices of the Royal Astronomical Society 181:375-389.

Gong, Kai., Hua. Liu & Ben-Long. Wang (2009), 'Water entry of a wedge based on SPH model with an improved boundary treatment', Journal of Hydrodynamics 21(6):750-757.

Grenier, N., M. Antuono, A. Colagrossi, D. Le Touze & B. Alessandrini (2009), 'An hamiltonian interface SPH formulation for multi-fluid and free surface flows', Journal of Computational Physics 228:8380-8393.

Grenier, N. & D. Le Touze (2008), 'An improved SPH method for multi-phase simulations', Proceedings of the 8nd international conference on hydrodynamics 11:2a-01.

Haskind, M.D (1946), 'The hydrodynamic theory of ship oscillations in rolling and pitching', Prikl. M at Mekh 10:33-66.

Hicks, D. L. & L. M. Liebrock (2000), 'Lanczos' generalized derivative: Insights and Applications', Applied Mathematics and Computation 112(1):63-73.

Hirdaris, S.E. & P. Temarel (2009), 'Hydroelasticity of ships: recent advances and future trends', Proceedings of the Institution of Mechanical Engineers 223(3):305.

Hirt, C.W., A.A. Amsden & J.L. Cook (1974), 'An arbitrary Lagrangian-Eulerian computing method for all flow speeds', Journal of Computational Physics 14:227-253.

Hockney, R.W. & J.W. Eastwood (1981), Computer simulation using particles, (New York: McGraw-Hill).

Hosseini, S. M., M.T. Manzari & S.K. Hannani (2007), 'A fully explicit three-step SPH algorithm for simulation of non-newtonian fluid flow', International Journal for Numerical Methods for Heat & Fluid Flow 17:715-735.

Hu, X. Y. & N. A. Adams (2006), 'A multi-phase SPH method for macroscopic and mesoscopic flows', Journal of Computational Physics 213(2):844-861.

Hu, X. Y. & N. A. Adams (2007), 'An incompressible multi-phase SPH method', J. Comput. Phys. 227(1):264-278.

Jenssen, C.B., T. Kvamsdal, K.M. Okstad & J. Amundsen (1998), 'Parallel methods for fluid-structure interaction', 98 proceedings of the 4th international workshop on applied parallel computing, large scale scientific and industrial problems.

Jia, Shuhui. (1987), Rigid Body Dynamics: Gao deng jiao yu chu ban she, China).

Jiang, Fangming, Mónica S. A. Oliveira & Antonio C. M. Sousa (2007), 'Mesoscale SPH modeling of fluid flow in isotropic porous media', Computer Physics Communications 176(7):471-480.

Johnson, G. R. & S. R. Beissel (1996a), 'normalized smoothing functions for SPH impact computations', International Journal for Numerical Methods in Engineering 39(16):2725-2541.

Johnson, G.R. (1994), 'Linking of Lagrangian particle methods to standard finite element methods for high velocity impact computations', Nuclear Engineering and Design 150:265-274.

Johnson, G.R., R.A. Stryk & S. R.Beissel (1996c), 'Interface effects for SPH impact computations', Structures under shock and impact IV:285-294.

Johnson, Gordon R., Robert A. Stryk & Stephen R. Beissel (1996b), 'SPH for high velocity impact computations', Computer Methods in Applied Mechanics and Engineering 139(1-4):347-373.

Joseph P. Morris, Patrick J. Fox, Yi. Zhu (1997), 'Modeling low reynolds number incompressible flows using SPH', Journal of Computational Physics 136:214-226.

Jun, Sukky, Wing Kam Liu & Ted Belytschko (1998), 'Explicit Reproducing Kernel Particle Methods for large deformation problems', International Journal for Numerical Methods in Engineering 41(1):137-166.

Khurram, Rooh.A. & Arif. Masud (2006), 'A multiscale/stabilized formulation of the incompressible Navier–Stokes equations for moving boundary flows and fluid–structure interaction', Computational Mechanics 38:403-416.

Korvin-Kroukovsky, B.V. (1955), 'Investigation of ship motions in regular waves', Trans. SNAME. 63.

Lanchester, W F. (1916), 'Torsional vibration of the tail of an airplane', &quot:; Reports and Memoranda V.

Lastiwka, Martin., Nathan. Quinlan & Mihai. Basa (2005), 'Adaptive particle distribution for smoothed particle hydrodynamics', International Journal for Numerical Methods in Fluids 47:1403-1409.

Lee, E. -S., C. Moulinec, R. Xu, D. Violeau, D. Laurence & P. Stansby (2008), 'Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method', J. Comput. Phys. 227(18):8417-8436.

Lee, E. S., D. Violeau & R. Issa (2010), 'Application of weakly compressible and truly incompressible SPH to 3-d water collapse in waterworks', Journal of Hydraulic Research 48:50-60.

Libersky, Larry D. & And A.G.Petschek (1991) 'Smoothed particle hydrodynamics with strength of materials', in, *in H.Trease, J. Fritts and W.Crowley (ed.): Proceeding of The Next Free Lagrange Confrence*: Springer Berlin/ Heidelberg): 248-257.

Liu, G.R. (2002), 'Mesh free methods: Moving beyond the finite element method', Chemical Rubber Boca Raton, FL.

Liu, G.R. & Y.T. Gu (2005), An introduction to meshfree methods and their programming, (Netherlands: Springer).

Liu, G.R. & M.B. Liu (2003b), Smoothed particle hydrodynamics: World Scientific Publishing Co. Pte.Ltd.).

Liu, M. B., G. R. Liu & K. Y. Lam (2003a), 'Constructing smoothing functions in smoothed particle hydrodynamics with applications', Journal of Computational and Applied Mathematics 155(2):263-284.

Liu, Wing Kam, Sukky Jun, Shaofan Li, Jonathan Adee & Ted Belytschko (1995b), 'Reproducing kernel particle methods for structural dynamics', International Journal for Numerical Methods in Engineering 38(10):1655-1679.

Liu, Wing Kam, Sukky Jun & Yi Fei Zhang (1995a), 'Reproducing kernel particle methods', International Journal for Numerical Methods in Fluids 20(8-9):1081-1106.

Lobovský, Libor & Jan Vimmr (2007), 'Smoothed particle hydrodynamics and finite volume modelling of incompressible fluid flow', Mathematics and Computers in Simulation 76(1-3):124-131.

Loon, R.V., P.D. Anderson, F.N. Van De. Vosse & S.J. Sherwin (2007), 'Comparison of various fluid-structure interaction methods for deformable bodies', Computer & Structures 85:833-843.

Lucy, L.B. (1977), 'Numerical approach to testing the fission hypothesis', Astronomical Journal 82:1013-1024.

Luo, Xiaojing., Jorg. Brunswig, Manuel. Manzke, Nils. Koliha, Earhan. Matin, Christian F. Janben & Thomas. Rung (2012) 'Analyses of Coupled Floating Bodies in Seaway', in David Le. Touze (ed), *2nd International Conference on Violent Flows* (Nantes, France: Publibook): 141-148.

Martin, J. C. & W. J. Moyce (1952), 'Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane', Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences 244(882):312-324.

Mattson, William. & Betsy M. Rice (1999), 'Near-Neighbor Calculations Using a Modified Cell-Linked List Method', Computer Physics Communications 119(2-3):135-148.

Mcbride, E.E. & L.J. Fisher (April 1953) 'Experimental investigation of the effect of rear-fuselage shape on ditching behaviour', in, *NACA Technical Note 2929* (Washington.

Moiseev, N.N. (1964), 'Introduction to the theory of osillations of liquid-containing bodies', Advances in Applied Mechanics 8:233-289.

Moiseev, N.N. & A.A. Petrov (1964), 'The calculation of free oscillation of a liquid in a motionless container', Advances in Applied Mechanics 9:91-154.

Monaghan, J. J. (1982), 'Why Particle Methods Work', SIAM Journal on Scientific and Statistical Computing 3(4):422-433.

Monaghan, J. J. (1989), 'On the problem of penetration in particle methods', Journal of Computational Physics 82(1):1-15.

Monaghan, J. J. (1992), 'SMOOTHED PARTICLE HYDRODYNAMICS', Annual Review of Astronomy and Astrophysics 30:543-574.

Monaghan, J. J. (1994), 'Simulating free surface flows with SPH', J. Comput. Phys. 110(2):399-406.

Monaghan, J. J. (1996), 'Gravity currents and solitary waves', Physica D: Nonlinear Phenomena 98(2-4):523-533.

Monaghan, J. J. (2002), 'SPH compressible turbulence', Monthly Notices of the Royal Astronomical Society 335(3):843-852.

Monaghan, J. J. & R. A. Gingold (1983), 'Shock simulation by the particle method SPH', Journal of Computational Physics 52(2):374-389.

Monaghan, J. J. & A. Kocharyan (1995), 'SPH simulation of multi-phase flow', Computer Physics Communications 87(1-2):225-235.

Monaghan, J. J. & A. Kos (1999), 'Solitary Waves on a Cretan Beach', Journal of Waterway, Port, Coastal, and Ocean Engineering 125(3):145-155.

Monaghan, J. J. & J.C. Lattanzio (1985b), 'A refined particle method for astrophysical problems', Astronomy and Astrophysics 149:135-143.

Monaghan, J.J. (1987) 'SPH meets the Shocks of Noh', in, *monash University Paper*.

Monaghan, J.J. (1988), 'An introduction to SPH', Computer Physics Communications 48:89-96.

Monaghan, J.J. (1992), 'Smoothed particle hydrodynamics', Ann. Rev. Astronom. Astrophys. 30:543-574.

Monaghan, J.J. (2001), 'Smoothed particle hydrodynamics code basics', Journal of the Korean Astronomical Society 34(4):203-207.

Monaghan, J.J. (2012), 'Smoothed Particle Hydrodynamics and Its Diverse Applications', Ann. Rev. Fluid Mech. 44:323-346.

Monaghan, J.J. & A. Kocharyan (1995), 'SPH simulation of multi-phase flow', Computer Physics Communications 87:225-235.

Monaghan, J.J. & J. Poinracic (1985a), 'Artificial viscosity for particle methods', Applied Numerical Mathematics 1:187-194.

Morris, J.P., P.J. Fox & Yi Zhu (1997), 'Modeling low reynolds number incompressible flows using SPH', J. Comput. Phys. 136:214-226.

Nikravesh, P.E. (1988), Computer-Aided Analysis of Mechanical Systems: Prentice-Hall, Englewood Cliffs, New Jersey).

Oger, G., M. Doring, B. Alessandrini & P. Ferrant (2006), 'Two-dimensional SPH simmulations of wedge water entries', Journal of Computational Physics 213:803-822.

Pozorski, J. & A. Wawrenczuk (2002), 'SPH computation of incompressible viscous flows', Journal of Theoretical Applied Mechanics 40:917.

Price, James F. (2006), 'Lagrangian and Eulerian representations of fluid flow: kinematics and the equations of motion', Woods Hole Oceanographics Institution, June 7.

Quinlan, N. J., M. Basa & M. Lastiwka (2006), 'Truncation error in mesh-free particle methods', International Journal for Numerical Methods in Engineering 66(13):2064-2085.

Rabczuk, T., S.P. Xiao & M. Sauer (2006), 'Coupling of meshfree methods with finite elements : basic concepts and test results', Communications in Numerical Methods in Engineering 22(10):1031-1065.

Rabzuk, T., S.P. Xiao & M. Saver (2000), 'Coupling of meshfree methods with finite elements: basic concepts and test results', Communications in Numerical Methods in Engineering 00:1-6.

Rafiee, A. & K.P. Thiagarajan (2008), 'Fluid-structure interaction imulation using an incompressible SPH method', ASME 27th international conference on offshore mechanics and arctic engineering:485-496.

Rafiee, Ashkan & Krish P. Thiagarajan (2009), 'An SPH projection method for simulating fluid-hypoelastic structure interaction', Computer Methods in Applied Mechanics and Engineering 198(33-36):2785-2795.

Randles, P. W. & L. D. Libersky (1996), 'Smoothed Particle Hydrodynamics: Some recent improvements and applications', Computer Methods in Applied Mechanics and Engineering 139(1-4):375-408.

Ritchie, Benedict W. & Peter A. Thomas (2001), 'Multiphase smoothed-particle hydrodynamics', Mon.Not.R.Astron.Soc 323:743-756.

Rugonyi, S. & K.J. Bathe (2001), 'On finite element analysis of fluid flows fully coupled with structural interactions', CMES 2:195-212.

Schussler, M. & D. Schmitt (1981), 'Comments on Smoothed Particle Hydrodynamics', Astro. Astrophys. 97:373-379.

Shao, Songdong & Edmond Y. M. Lo (2003), 'Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface', Advances in Water Resources 26(7):787-800.

Shao, Songdong. (2009), 'Incompressible SPH simulation of water entry of a free-falling object', International Journal for Numerical Methods in Fluids 59:91-115.

Stellingwerf, Rf. & Ca. Wingate (1994), 'Impact modelling with SPH', Memorie della societa astronomia italiana 65:1117.

Streckwall, H., O. Lindenau & L. Bensch (2007), 'Aircraft ditching: a free surface / free motion problem', Archives of Civil and Mechanical Engineering:177-190.

Sun, F., M. Tan & T. J. Xing (2011), 'Investigations of boundary treatments in incompressible smoothed particle hydrodynamics for fluid-structure interactions', Proceedings of the 2nd international conference on fluid mechanics and heat and mass transfer 2011:92-97.

Sun, Fanfan., Mingyi. Tan & Jing Tang. Xing (2012) 'Air-water two phase flow simulation using smoothed particle hydrodynamics', in David Le Touze (ed), *2nd International Conference on Violent Flows* (Nantes, France: Publibook): 58-63.

Sussman, M., P. Smereka & S. Osher (1994), 'A level set approach for computing solutions to incompressible two-phase flow', Journal of Computational Physics 114:146-159.

Theodorsen, T. (1934) 'General theory of aerodynamic in stability and themechanism of flutter', in, *NACA TR* (In : Twentieth Annaul Report of the National Advisory Committee for Aerodynamics 1934, US Government Printing office, Washington (1935): 413-433.

Toso, Nathalie Renée Solange. (2009) 'Contribution to the modelling and simulation of aircraft structures impacting on water', in, *Faulty of Aerospace Engineering and Geodesy of the University Stuttgart* Stuttgart University).

Tveitnes, T., A.C. Fairlie-Clarke & K. Varyani (2008), 'An experimental investigation into the constant velocity water entry of wedge-shaped sections', Ocean Engineering 35(14-15):1463-1478.

Viccione, G., V. Bovolin & E. Pugliese Carratelli (2008), 'Defining and optimizing algorithm for neighbouring particle identification in SPH fluid simulations', International Journal for Numerical Methods in Fluids 58:625-638.

Vorst, H.A. Van Der (1992), 'BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetri linear systems', Society for Industrial and Applied Mathematics 13:631-644.

Westergaard, H.M. (1933), 'Water pressure on dams during earthquakes', Trans Amer. Soc. Civ. Eng. 98:418-433.

Wróblewski, Pawel., Marius. Z Kopeć & Krzysztof. Boryczko (2007), 'SPH – A comparison of neighbor search methods based on constant number of neighbours and constant cut-off radius', TASK QUARTERLY 11 NO 3:273-283.

Xing, J. & W. Price (1991), 'A mixed finite element method for the dynamic analysis of coupled fluid-solid interaction problems', Proceedings of the Royal Society of London A433:235-255.

Xing, J. T., W.G. Price & Y. G. Chen (2003), 'A mixed finite-element finite-difference method for nonlinear fluid-structure interaction dynamics. I. Fluid-rigid structure interaction', The Royal Society 459:2399-2430.

Xing, J.T., W. Price & Q. Du (1996), 'Mixed finite element substructure-subdomain methods for the dynamical analusis of coupled fluid-solid interaction problems', Philosophical Transactions: Mathematical, Physical and Engineering Sciences 354:259-295.

Xing, Jing Tang., Sheng. Zhou & Erjie. Cui (1997), 'Oversview of fluid structure interaction mechanics', Advances in Mechanics 27:19-38.

Yao, Zhenhua., Jiansheng. Wang & Min. Chen (2004), 'Improved O(N) neighbor list method using domain decomposition and data sorting', High performance computation for engineered systems.

Zhang, G.M. & R.C. Batra (2004), 'Modified smoothed particle hydrodynamics method and its application to transient problems', Computational Mechanics 34:137-146.

Zhang, Tao., Shu. Li & Dai Heng. Chao (2012), 'The suction force effect analysis of large civil aircraft ditching', Science China 55:2789-2797.

Zhao, R., O. Faltinsen & J. Aarsnes (1997), 'Water entry of arbitrary two-dimensional sections with and without flow separation', 21st Symposium on Naval Hydrodynamics.

Zhou, Z.Q., J.O. Dekat & B. Bunchner (1999), 'A nonlinear 3-d approach to simulate green water dynamics on deck'.

Publication list

Sun, F., M. Tan & T. J. Xing (2011), 'Investigations of boundary treatments in incompressible smoothed particle hydrodynamics for fluid-structure interactions', Proceedings of the 2nd international conference on fluid mechanics and heat and mass transfer 2011:92-97.

Sun, Fanfan., Mingyi. Tan & Jing Tang. Xing (2012) 'Air-water two phase flow simulation using smoothed particle hydrodynamics', in David Le Touze (ed), *2nd International Conference on Violent Flows* (Nantes, France: Publibook): 58-63.

Sun, F., M. Tan & T. J. Xing (2013), 'Application of incompressible smoothed particle hydrodynamics method for 3d fluid solid interaction problem', Proceedings of the 10th international conference on fluid mechanics 2013:92-97.

Sun, F., M. Tan & T. J. Xing (2012), 'Efficient implementation of boundary conditions for rigid structure-incompressible fluid interactions using SPH method', summitted to International Journal of Mechanical Sciences.