Development of a full particle PIC method for simulating nonlinear wave-structure interaction

Chen, Qiang

Award date:
2017

Awarding institution:
University of Bath

Development of A Full Particle PIC Method for Simulating Nonlinear Wave-Structure Interaction

Qiang Chen

A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Architecture and Civil Engineering

February 2017

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First of all, I acknowledge with thanks the financial support of University of Bath and HR Wallingford for this research.

I would like to sincerely thank my academic supervisors Dr. Jun Zang and Dr. Chris Williams at University of Bath, and industrial supervisors Dr. David Kelly and Dr. Aggelos Dimakopoulos at HR Wallingford, for all their help on this research. Particularly, great thanks go to Dr. Jun Zang for her continuous offering of guidances, supports and chances to not only my work but also my life in UK throughout my PhD study. I thank Dr. David Kelly (now at Florida International University, USA) for his patient teaching on scientific computing and my English writing, and his great initial guidance on this research. I thank Dr. Aggelos Dimakopoulos for his many thought-provoking comments on the research, help on my scientific writing and introducing me to many experts at HR Wallingford. I thank Dr. Chris Williams for his many constructive suggestions on the research. I thank Dr. Chris Blenkinsopp at University of Bath and Prof. Qingwei Ma at City University of London for their time dedicated to carefully reading this thesis and their helpful comments on improving the quality of this thesis. Great thanks also go to Prof. Bin Teng and Prof. Dezhi Ning at Dalian University of Technology, China, for their recommendations for my PhD study with Dr. Jun Zang and some helpful discussions on the research.

I also thank Dr. Alison Evans and Mr. Eoghan Gannon at University of Bath, and Dr. Giovanni Cuomo at HR Wallingford, for their supports on this PhD studentship. The use of the Balena High Performance Computing Service at the University of Bath and Dr. Jun Zang's premium user account for this service are appreciated.

Many thanks go to all my friends and colleagues at the Research Unit for Water, Environment and Infrastructure Resilience (WEIR) of University of Bath, particularly Dr. Liang Sun, Dr. Lifen Chen and Dr. Jinxuan Li, for their accompanying and help. I also thank Dr. Jeremy Spearman at HR Wallingford for his comments and contributions on the introduction of fallpipe rock dumping tests.

Finally, I sincerely thank my parents for their endless love and support, and I am grateful to my wife Mrs Zhi Liang for her love and accompanying. I could not have finished this thesis without their supports.

PUBLICATIONS

The following publications have been produced as parts of this thesis:

Journal articles:

Chen, Q., Zang, J., Kelly, D. M. and Dimakopoulos, A. S., 2017. A 3D parallel Particle-In-Cell solver for wave interaction with vertical cylinders. *Submitted*.

Chen, Q., Zang, J., Kelly, D. M. and Dimakopoulos, A. S., 2017. A 3D numerical study of solitary wave interaction with vertical cylinders using a parallelised Particle-In-Cell solver. *Accepted by Journal of Hydrodynamics*.

Chen, Q., Kelly, D. M., Dimakopoulos, A. S. and Zang, J., 2016. Validation of the PICIN solver for 2D coastal flows. *Coastal Engineering*, 112, pp. 87-98.

Chen, Q., Zang, J., Dimakopoulos, A. S., Kelly, D. M. and Williams, C. J. K., 2016. A Cartesian cut cell based two-way strong fluid-solid coupling algorithm for 2D floating bodies. *Journal of Fluids and Structures*, 62, pp. 252-271.

Kelly, D. M., **Chen, Q.** and Zang, J., 2015. PICIN: a particle-in-cell solver for incompressible free surface flows with two-way fluid-solid coupling. *SIAM Journal on Scientific Computing*, 37 (3), B403-B424.

Conference papers:

Chen, Q., Zang, J., Kelly, D. M. and Dimakopoulos, A. S., 2017. A 3D numerical study of solitary wave interaction with vertical cylinders using a parallelised Particle-In-Cell solver. *In: 32th International Workshop on Water Waves and Floating Bodies*, China.

Chen, Q., Kelly, D. M., Spearman, J., Dimakopoulos, A., Zang, J. and Williams, C., 2015. CFD Modelling of Fall Pipe Rock Dumping Using PICIN. *In: Coastal Sediments 2015*, USA.

Chen, Q., Zang, J., Kelly, D., Williams, C. and Dimakopoulos, A., 2015. Particle-In-Cell Numerical Solver for Free Surface Flows with Fluid-Solid Interactions. *In: 30th International Workshop on Water Waves and Floating Bodies*, UK.

SUMMARY

During the past few decades, Computational Fluid Dynamics (CFD) modelling has become very popular in the coastal and offshore engineering community. Both Eulerian and Lagrangian methods have achieved great successes; typical examples are the grid-based OpenFOAM® model and the meshless Smoothed Particle Hydrodynamics (SPH) method based model (e.g. SPHysics). While the former tends to be more efficient and has advantages in enforcing incompressibility and boundary conditions via use of a grid, the latter is more suitable for handling large free-surface deformations using particles. In an attempt to combine the advantages of both methods, the Particle-In-Cell (PIC) method was devised through a combined use of particles and grid. However, so far this hybrid method has not been very well exploited for use in the coastal and offshore engineering field, where modelling complex wave-structure interaction with computational efficiency still remains an important challenge.

This thesis develops a novel "full particle" PIC based numerical model that solves the incompressible Newtonian Navier-Stokes equations for single-phase free-surface flows with an emphasis on fluid-structure interaction. The use of the phrase "full particle" here indicates that all of the fluid properties, such as the mass and momentum, are assigned only to the particles, rather than being split between the particles and grid as is the case in "classical" PIC. The novelty of the model lies in the fact that the particles are employed to solve the nonlinear advection term and track the fluid configuration (including the free surface), while the underlying grid is solely used for computational convenience for solving the non-advection terms. In addition, a tailored Distributed Lagrange Multiplier method and a Cartesian cut cell based two-way strong coupling algorithm are incorporated for fluid-structure interaction. The model is developed in both two and three spatial dimensions, and the 3D model is parallelised using the Message Passing Interface (MPI) approach. The model is validated using benchmark tests in the coastal and offshore engineering field with simulating nonlinear wave-structure interaction being the principal interest. It is shown that the present "full particle" PIC model is flexible, efficient (in terms of CPU cost) and accurate when modelling complex free-surface flows and the violent interaction of such flows with (surface-piercing) structures of arbitrary shape and degree of freedom. With new innovations, the model has great potential to become a high quality numerical tool for use in coastal and offshore engineering applications.

LIST OF ABBREVIATIONS

1D, 2D, 3D	one dimensional, two dimensional, three dimensional
BCG	bi-conjugate gradient
CFD	computational fluid dynamics
CIP	constrained interpolation profile
DEM	discrete element method
DLM	distributed lagrange multiplier
DoF	degree of freedoms
FLIP	fluid-implicit-particle algorithm
FVM	finite volume method
HPCS	high performance computing systems
IB	immersed boundary
ISPH	incompressible SPH
LCS	low-crested structure
LHS	left hand side
MAC	Marker-And-Cell
MLS	moving least square
MPI	message passing interface
NS	Navier-Stokes
NWT	numerical wave tank
PFEM	particle finite element method
PIC	Particle-In-Cell
PISO	pressure implicit with splitting of operator
PPE	pressure Poisson equation
RANS	Reynolds-averaged Navier Stokes
RHS	right hand side
RMSE	root mean square errors
SDF	signed distance function
SIMPLE	semi-implicit method for pressure linked equations
SPH	smoothed particle hydrodynamics
VOF	volume of fluid
WCSPH	weakly compressible SPH
WENO	weighted essentially non-oscillatory

LIST OF NOTATIONS

The major notations used in this thesis are listed:

Chapter 3

$\boldsymbol{u}$	general velocity field; general velocity vector on the grid
$\tilde{\boldsymbol{u}}$	tentative velocity on the grid
$\boldsymbol{u}^n$	grid velocity at n time-steps
$u_{i+1/2,j}$	horizontal velocity at the centre of a cell edge $(i+1/2, j)$
$w_{i,j+1/2}$	vertical velocity at the centre of a cell edge $(i, j+1/2)$
$\boldsymbol{a}$	velocity change on the grid, $\boldsymbol{a}^{n+1} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^n$
$\boldsymbol{r}$	general position vector
$\boldsymbol{X}$	general position vector for grid
$\boldsymbol{x}$	general position vector for particle
$\boldsymbol{v}_p$	velocity vector for particle
m_p	particle mass
t	time
Δt_n	time step at nth time-step
Δx, Δz	grid size in x- and z-direction
Ω	overall computational domain
$\partial\Omega$	boundary of overall computational domain
$\zeta(\boldsymbol{r}, t)$	free-surface boundary
p	pressure
$\boldsymbol{f}$	external force term
ρ	fluid density
ν	kinematic viscosity of a fluid
c	an empirical blending coefficient
ϕ	singed distance funtion with respect to free surface
W_i	kernel function (subscript i denotes the reference point index)
k_h	kernel length
σ_{pg}	normalised kernel function
u_{ext}	extrapolated velocity component

N_r	time-steps interval for the redistribution scheme
C_F	the Courant number

Chapter 4

Ω_f	fluid phase
Ω_s	structure phase
$\partial\Omega_s$	structure interface
$\boldsymbol{\Pi}$	extra stress tensor in addition to the pressure
ρ_S	structure density
$\boldsymbol{u}^{n+1/2}$	divergence-free velocity field after solving solids as if they were fluids
$\dot{\boldsymbol{u}}$	grid velocity within the structure phase after the tentative advection step
$\hat{\boldsymbol{u}}$	grid velocity within the structure phase after the correction step
$\hat{\boldsymbol{u}}_R$	final grid velocity within the structure phase
θ	volume fraction of a velocity cell that is occupied by structures
ϵ_p, ϵ_w	positive stiffness coefficient
R	radius of a circular particle
λ	distance range
$\boldsymbol{U}$	structure veloicty
$\boldsymbol{Y}$	structure displacement
$\boldsymbol{f}_{co}$	overall contact force between structures

Chapter 5

$\boldsymbol{U}_b$	velocity field imposed on the structure boundary
$\boldsymbol{U}$	structure veloicty
ϕ_s	signed distance funtion with respect to the structure interface
G_{ij}	cell $(i-1/2, i+1/2) \times (j-1/2, j+1/2)$
$\boldsymbol{n}$	unit normal vector
$E_{i+1/2,j}$	fraction of grid face $(i+1/2, j)$ that is open to water
$V_{i+1/2,j}$	cell volume of velocity cells (centred at $u_{i+1/2,j}$) occupied by structures
Δl	length of line segment representing structure boundary
$\boldsymbol{Q}_k$	structure boundary velocity transferred from the pressure immediately surrounding the structure

β	fraction ratio of the associated cell edge that is occupied by the structures
$\xi(t)$	location of wave paddle
h	water depth
H	wave height
ω	wave frequency (in rad)
T	wave period
L	wave length
k	wave number
η	free-surface elevation
ξ_0	initial position of the wave crest
D	diameter of cylinder
D_L	the absorption zone length

CHAPTER 1

INTRODUCTION

With increasing computing power, Computational Fluid Dynamics (CFD) modelling has been developed in many research areas for more than half a century. Both two dimensional (2D) and three dimensional (3D) simulations have become possible and increasingly accurate and efficient. In the coastal and offshore engineering field, the Navier-Stokes (NS) equations have been extensively solved using Eulerian methods, Lagrangian methods and hybrid Eulerian-Lagrangian methods. Whilst pure Eulerian methods have become popular due to their compelling abilities in enforcing incompressibility and boundary conditions and their relatively high computational efficiency, pure Lagrangian (meshless) approaches are popular due to their flexibilities in handling complex free-surface flows such as wave breaking, slamming and overturning. A successful example is the Smoothed Particle Hydrodynamics (SPH) method, of which, however, the main drawback is the highly demanding CPU cost. In this sense, hybrid methods are appealing as they have the chance to combine the advantages and mitigate the disadvantages of the pure methods. This thesis is thus motivated by developing a hybrid numerical method for simulating complex free-surface flows and the interaction of such flows with coastal and offshore structures. In particular, this is based on the Particle-In-Cell (PIC) method where both particles and a grid are utilised. While the particles are used for tracking the free surface and solving the nonlinear advection term in a Lagrangian manner, the underlying grid is employed for computational convenience for solving the non-advection terms in an Eulerian sense. The idea being that the numerical method should have both the flexibility of pure Lagrangian methods and the efficiency of pure Eulerian methods. The following sections first introduce the PIC methods, followed by a brief introduction of relevant numerical works in the coastal and offshore engineering field using other related numerical approaches. The aim and scope of this thesis are then detailed, and the structure of this thesis is finally introduced.

1.1 Particle-In-Cell methods

1.1.1 Background and development history

The PIC approach was originally devised for hydrodynamic problems at the Los Alamos National Laboratory in the 1950's (Harlow, 1955; Evans and Harlow, 1957; Harlow, 1957; Harlow et al., 1959). This method adopts a double grid system: Eulerian grid and Lagrangian particles. Fluid quantities such as velocity, density and pressure are stored in cells, while mass is stored in particles. The particles can move through the grid, and thus transfer mass and momentum between cells. According to the PIC methodology, the computations are usually divided into several phases. In the first phase, the governing equations, ignoring the advection terms, are solved on the grid. The remaining advection terms are then handled in the second phase by moving the particles in a Lagrangian manner. The third phase accomplishes the update of fluid quantities on the grid according to the transportation via the particles and prepares for the next cycle of computation. It is noteworthy that the fluid velocity is interpolated back and forth between the grid and particles to move the particles and hence transport fluid quantities. Through a combined use of particles and grid, the PIC method was devised in an attempt to mitigate the diffusion due to fluid discontinuities in traditional Eulerian methods, and grid tangling due to large distortions in traditional Lagrangian (mesh-based) methods (Harlow, 1988). As a result of these advantages, the PIC method was applied to a variety of complicated problems shortly after its invention. These include simulations of shock waves and viscous flows (see Harlow et al. (1959)), and supersonic flows (see Evans and Harlow (1958)). Later, Harlow (1964) further developed this approach for practical usage and this version is usually termed the "classical" PIC method.

A typical characteristic of the "classical" PIC method is that the velocity is interpolated back and forth between particles and grid at each time step, which leads to large numerical diffusion. Additionally, the "classical" PIC method also suffers from problems such as being highly demanding in terms of storage, and very noisy due to the discontinuity of mass and momentum transfer when the number of particles is small (Nishiguchi and Yabe, 1983; Harlow, 2004). Thus, many attempts have been made to improve this method.

One branch of development resulted in the "full particle" PIC method, assigning all fluid properties such as mass and momentum only to the particles, in order to reduce the numerical dissipation. This is inspired by the fact that mass is perfectly conserved in the "classical" PIC method as it is carried by the particles (Brackbill et al., 1988). The grid here, by contrast, is solely used for computational convenience. That is, computing acceleration (i.e. the change of velocity) based on the non-advection terms of the governing equations. These velocity changes are then interpolated back to update the particle velocity. Since there is no direct transfer of velocity between particles and grid, and the velocity change is relatively small, the numerical diffusion is significantly reduced. Also, the numerical noise caused by using the finite numbers of fluid particles in the "classical" PIC method is suppressed as the particles are not used to transfer quantities between grid. Typical examples of the models that employ the "full particle" PIC framework are the GAP (grid and particles) solver (Marder, 1975) and the FLIP (fluid-implicit-particle) solver (Brackbill and Ruppel, 1986; Brackbill et al., 1988). In addition, it is worth mentioning that the "full particle" PIC methods are widely used in the plasma research community (see e.g. Leboeuf et al. (1979); Birdsall and Langdon (1985); Eastwood (1986)). Nevertheless, the "full particle" PIC method also suffers from a few difficulties; these are discussed in the following section.

Another direction of research is based on improving the "classical" PIC method. For example, Nishiguchi and Yabe (1982, 1983) proposed the SOAP scheme which uses a high-order scheme for energy and momentum advection in the "classical" PIC framework. The idea is to first give each particle a finite area, with the particles located at the centres of the areas. Before advecting the particles, the distribution of local cellwise quantity is reproduced within the predefined finite areas. Then, after moving the particles, and the associated finite areas, the quantity in a grid cell is updated via integrating over the overlapped areas of the cell and the finite areas of the particles nearby, where the distribution of a quantity has already been defined. As the energy and momentum transport becomes a much more continuous function of the particle displacement, this approach improves the accuracy and reduces the numerical viscosity and noise of the "classical" PIC method. As another example, Jiang et al. (2015) believe that the primary reason for the numerical dissipation in "classical" PIC is that the angular momentum is not conserved when the velocity is interpolated from grid to particles. By storing the lost angular momentum on the particles when conducting the grid-to-particle velocity interpolation, and redis-

tributing it back when conducting the particle-to-grid interpolation, they managed to significantly reduce the numerical diffusion and retain the stability of the "classical" PIC method. Nevertheless, due to the "classical" PIC nature, their method is inferior to the "full particle" PIC based algorithm FLIP in preserving energy for some test cases presented.

In addition to the above-mentioned developments, there are some other variants of the PIC method. While the original PIC method is designed for compressible flows, Harlow and Welch (1965) extended the method to solve incompressible free-surface flows, leading to the Marker-and-Cell (MAC) method. In the MAC method, the particles are employed only to indicate the flow configurations; they do not participate in the calculation as do the particles in the PIC method. Hirt and Shannon (1968) and Nichols and Hirt (1971) later developed the MAC method with an improved implementation of the free-surface boundary conditions. Viecelli (1971) employed the MAC method to model the interaction of incompressible flows with wall boundaries of arbitrary shapes. Whilst the MAC method becomes more Eulerian compared with the PIC method, another variant, the Particle-and-Force (PAF) method (Daly et al., 1965) takes the opposite direction (Harlow, 2004), i.e. being more Lagrangian. This PAF approach eliminates the use of an underlying grid, and puts all of the calculations onto the particles themselves. The particles carry the fluid properties that are updated through the inter-particle forces corresponding with fluid mechanics. However, this method was not further investigated due to the success of other approaches of the same type at that time (Harlow, 2004).

1.1.2 Difficulties and solutions of modern PIC methods

Modern PIC methods have certain limitations and some disadvantages that require further investigations. One is the difficulty in achieving numerical stability while being free of unwanted numerical dissipation. For example, in "full particle" PIC, despite the great success in reducing the numerical dissipation, there is a potential risk of instability in the velocity field. This is because in "full particle" PIC the velocity field carried by the particles is incremented through the velocity change on the grid, rather than being dissipated or filtered naturally through direct interpolation from the grid as is the case in "classical" PIC. Therefore, the errors in incrementing the particle velocity can accumulate uncontrollably during the simulation (Jiang

et al., 2015). To make a compromise between stability and accuracy, Zhu and Bridson (2005) suggested using a blending between "classical" PIC and "full particle" PIC. That is, when transferring the velocity field from grid to particles, a weighted average is implemented between the velocity itself ("classical" PIC) and the velocity change ("full particle" PIC). In fact, the weight coefficient could be used to tune the desired numerical viscosity (Zhu and Bridson, 2005). Of course, for modelling water type problems where fluid viscosity is small, "full particle" PIC is preferable. For example, Ando et al. (2012) used a weight coefficient 0.95 for the "full particle" PIC part and only 0.05 for the "classical" PIC part.

Another problem inherent in the PIC methods is particle clumping and uneven spatial particle distributions over time as observed in, for example, Brackbill (1988); Ando et al. (2012); Edwards and Bridson (2012) and Wang et al. (2015). The irregular particle distributions may lead not only to low accuracy in data transfer between particles and grid but also to unphysical voids in the fluid region. A number of reasons for this problem and their corresponding solutions are reviewed herein. One reason may be simply the collisionless nature of particle motion in PIC. The particles are normally moved through the grid velocity, and thus, the truncation errors existing in this process could lead to inaccurate movements of the particles and hence irregular particle distributions, especially for certain instances where a large velocity gradient occurs. Solutions to this issue are to use high-order interpolation schemes for the velocity transfer (Edwards and Bridson, 2012) and high-order methods for advecting the particles, such as those from the Runge-Kutta family (Ralston, 1962). Another factor causing particle disordering is the so-called ringing instability as reported in Brackbill (1988, 2005). The ringing instability is a numerical instability due to aliasing errors, as there are usually more particles than grid cells and hence some of the higher frequency components of the motion at the particle level may not be represented at the grid level. The aliases introduce additional resonances that could lead to instability through a nonlinear interaction (Brackbill, 1988). Nevertheless, it is reported in the same work that the strength of ringing instabilities can be reduced by using a high-order interpolation scheme for data transfer between particles and grid and be controlled via an implicit time differencing scheme. A third reason causing the particle disordering, as suggested by Wang et al. (2015), is an upset of the divergence-free condition when transferring the velocity field from grid to particles to solve the transport term. Based on this explanation, Wang et al. (2015) proposed a divergence-free interpolation scheme and the results are very promising in reducing

the particle clustering. Finally, in addition to the explanations and solutions mentioned above, it is worth mentioning that Ando et al. (2012) proposed an anisotropic position correction scheme that directly shifts particles in space when particles get too close to each each. This idea is also shared by other particle-based methods such as SPH (see, e.g., Lind et al. (2012) and Colagrossi et al. (2012)). Furthermore, Edwards and Bridson (2012) suggested a particle reseeding method that adds and deletes particles in cells approaching too few or too many particles, respectively.

PIC methods are commonly reported to be of low-order accuracy (Edwards and Bridson, 2012). One reason for this is the low-order accuracy in the information transfer between scattered particles and the uniform grid. While the grid-to-particle interpolation is somewhat straightforward and can employ high-order schemes such as the weighted essentially non-oscillatory (WENO) scheme proposed by Edwards and Bridson (2012), the particle-to-grid interpolation is usually of low-order accuracy due to the uneven particle distributions as mentioned above. Edwards and Bridson (2012) proposed using the moving least square (MLS) method for the particle-to-grid interpolation and the fourth-order WENO scheme for the grid-to-particle interpolation; their method achieves results of very high-order accuracy. However, only unconstrained dynamics such as the solution of shallow water equations are considered in their work. More recently, Edwards (2015) extended this method to constrained dynamics and solved the NS equations. It is nevertheless unclear in that paper whether the extended method retains a high-order accuracy for simulating free-surface flows, and the interactions of such flows with structures, which still remains a challenge for modern PIC methods.

PIC methods are generally very demanding in memory storage due to the double grid system (i.e. particles and grid), which probably has limited its applications in the early days. In this thesis, the proposed PIC based model is parallelised through the Message Passing Interface (MPI) approach for the simulations in three spatial dimensions. It is demonstrated that the MPI parallelisation enables the present PIC model to simulate large-scale 3D problems (e.g. those require a large number of grid cells and particles).

1.1.3 Prospects of modern PIC methods

Despite the difficulties introduced above, the PIC methods have also shown certain advantages in computing fluid dynamics. The particles, representing either partially ("classical" PIC) or fully ("full particle" PIC) the fluid domain, are effectively used for solving the transport terms in governing equations and tracking flow evolution. This is computationally efficient and leads to much less artificial diffusion and numerical complexities (Bridson, 2008), compared with pure Eulerian methods where equation-based calculations pose certain difficulties when simulating fluid flows with large free-surface distortions. Meanwhile, PIC retains the utilisation of a grid for calculation of non-advection terms. The grid is able to discretise governing equations in a natural manner, particularly for calculating gradients, thus facilitating the enforcement of boundary conditions and the incompressibility condition. Through implementing the main calculations on the grid, the PIC methods can easily employ optimised schemes that are available in the literature for high efficiency in terms of CPU cost. Furthermore, compared to pure Lagrangian (meshless) methods, the use of a grid requires a much smaller number of neighbours to calculate numerical gradients, and there is no requirement for neighbour searching, which by contrast is very costly in pure Lagrangian (meshless) methods such as SPH. Therefore, the PIC methods have access to the advantages of both Eulerian and Lagrangian methods through the joint use of particles and grid, and hence have great potential for applications in complex hydrodynamic problems.

1.2 Relevant CFD works on fluid-structure interactions

The topic of fluid-structure interaction which includes, amongst other things, wave generation and absorption, wave slamming, green water overtopping and floating structures has been widely studied both experimentally and numerically in the coastal and offshore engineering field (Faltinsen et al., 2004; Chen et al., 2014b; Gao and Zang, 2014; Oliveira et al., 2012; Zhao and Hu, 2012). CFD models have become increasingly popular in this field as an efficient tool for physical process understanding and structure optimisation. This section lists some of these relevant models based on pure Eulerian methods, pure Lagrangian (meshless) methods and hybrid methods.

1.2.1 Eulerian methods

As a representative in this group, the Eulerian Volume of Fluid (VOF) based solver interFOAM, from the open-source OpenFOAM® modelling suite, has become a very popular numerical tool for investigations in this area. Jacobsen et al. (2012) implemented wave generation and absorption in interFOAM using the wave relaxation zone concept and used the resulting code to investigate wave propagation and breaking. Dimakopoulos et al. (2016) further optimised this toolkit and demonstrated that it is capable of modelling random and directional waves. Higuera et al. (2013a,b) developed and validated their OpenFOAM® tool for realistic wave generation and active absorption. The results of their model show a good agreement with experimental and other numerical data in terms of coastal engineering processes of wave breaking, run up and undertow currents. Chen et al. (2014b) enhanced the OpenFOAM® modules used for wave generation and absorption and subsequently investigated extreme wave interaction with a vertical cylinder. Using the same model Chen et al. (2014a) further investigated wave interaction with a rectangular box having only roll motion. Gao and Zang (2014) solved the Reynolds-averaged Navier Stokes (RANS) equations with $k-\omega$ turbulence model using OpenFOAM® in order to study different types of breaking wave impact on a vertical wall.

1.2.2 Meshless Lagrangian methods

In terms of pure Lagrangian (meshless) methods developed for the coastal and offshore engineering field, the SPH method has become very popular and successful during the past decade (Violeau and Rogers, 2016). The basic idea behind SPH is that, at the particle level, any variable A of the fluid domain can be expressed as an average of the neighbouring particles with a kernel function:

$$A(\boldsymbol{r}) = \sum \frac{m_a}{\rho_a} A_a W(\boldsymbol{r} - \boldsymbol{r}_a, h) \ , \tag{1.1}$$

where W is the kernel function; h is the so-called kernel length determining the neighbourhood of particles; $\boldsymbol{r}$ is the position vector; m_a and ρ_a are the mass and density of particle a, respectively. With Equation 1.1, the gradient or Laplace operator of variables can be transferred onto the operation on the kernel function, and in

such manner the NS equations can be expressed in a SPH form and solved (see e.g. Liu and Liu (2003)). A wide variety of SPH applications can be found in the literature. Oger et al. (2006) studied a rigid body impacting a water surface, and thus demonstrated the ability of SPH to capture the large pressure variation observed during the impact. Rogers et al. (2010) applied the SPH approach to an engineering problem involving caisson breakwater movement; their results agree well with the experimental data. Bouscasse et al. (2013) developed a fully coupled fluid-solid interaction algorithm based on SPH for the simulations of nonlinear fluid-structure interaction including, for example, wave packet action on a 2D floating box.

Other meshless methods for coastal and offshore applications have also been developed. For example, Koshizuka et al. (1998) used the Moving Particle semi-implicit (MPS) method to study wave breaking and its interaction with a floating body. Ma (2005a,b) extended the Meshless Local Petrov-Galerkin (MLPG) method, and further developed it based on Rankine source solution (named as MLPG_R method), to simulate nonlinear water waves. Zhou (2010) applied the MLPG_R method to simulate breaking waves and the interaction of such waves with fixed structures.

1.2.3 Hybrid methods

Hybrid Eulerian-Lagrangian methods are attractive for use in the coastal and offshore engineering applications as they attempt to combine the advantages of both Eulerian and Lagrangian methods. A well-known hybrid method is the arbitrary Lagrangian-Eulerian (ALE) approach, where the mesh inside the computational domain can move arbitrarily to best accompany the motions of fluid and boundaries in order to get the best features of both Lagrangian and Eulerian methods. The ALE model was first introduced to fluid dynamic problems by Hirt et al. (1974), where a methodology based on a movable finite difference mesh was presented. Ramaswamy (1990) developed an ALE model for incompressible viscous free-surface flows based on the finite element method. Zhou and Stansby (1999) developed an ALE model for simulating wave behaviour over bars.

In contrast, the hybrid PIC methods have been applied in the coastal and offshore engineering field only very recently; very few relevant applications are found in the

literature. Kelly (2012) developed a "full particle" PIC solver and employed it for simulating solitary wave propagation and breaking over an idealised beach in two spatial dimensions; this initial work shows that the PIC methods are re-emerging with new improvements (Violeau and Rogers, 2016). Very recently, Maljaars (2016) simulated a similar 2D solitary wave case to that studied in Kelly (2012); the author used a PIC based model that incorporates a finite element discretisation of the governing equations at the background grid. Yet, in these papers, the complex fluid-structure interactions widely encountered in the coastal and offshore engineering field such as wave overtopping, wave impact and floating structures are not further considered, nor are the properties of the PIC methods for these kinds of applications comprehensively understood.

1.3 Aims and scope

Originally designed as a numerical tool for the calculation of fluid dynamics, PIC methods have been applied to many other research areas such as, as mentioned above, plasma simulation, solid mechanics and computer graphics. It is, however, noted by this thesis that the PIC methods have not attracted sufficient attention from both industry and academics in the coastal and offshore engineering area, where the simulation of complex scenarios involving violent interactions of free-surface waves and maritime structures is still very challenging. The potential capability of PIC methods in handling large free-surface deformations while being computationally efficient may be ideal to serve this community. This thesis thus focuses on this topic and introduces a novel "full particle" PIC solver that is able to handle both one-way and two-way fluid-structure interactions in complex coastal and offshore environments. The performance and capability of the proposed model are demonstrated through the examination of a number of benchmark tests in this area.

1.4 Thesis structure

The work of this thesis is divided into seven chapters.

In chapter 1, the motivation, aim and scope of this thesis, as well as a review of PIC methods are introduced.

In chapter 2, a literature review on different subjects in the development of a numerical solver is given. Attention is focused on techniques for several main aspects: 1) solving the nonlinear advection term in NS equations using Eulerian methods, Lagrangian methods and coupled methods; 2) interpolation schemes for information transfer between scattered particles and a uniform grid; 3) techniques for tracking or capturing free surfaces of fluid flows; 4) numerical schemes for solving the NS equations; 5) numerical approaches for fluid-structure interactions based on Eulerian grid.

In chapter 3, the detailed methodology used in the proposed "full particle" PIC model for free-surface flows is described, without the fluid-structure interaction component which is developed in the following chapters. With the techniques described in this chapter, a numerical wave tank is established. The test cases of a standing wave and dam break are used to test the model for simulating free-surface flows.

In chapter 4, the details of developing a tailored Distributed Lagrange Multiplier method for fluid-structure interaction within the "full particle" PIC framework developed in chapter 3 are introduced. The application in this chapter is focused on the simulation of fall-pipe rock dumping.

In chapter 5, the details of developing a cut cell based two-way strong coupling algorithm for fluid-structure interactions within the "full particle" PIC framework developed in chapter 3 are introduced. A numerical wave tank with wave generation and absorption is established. The test cases used in this chapter are wave interaction with fixed and movable structures typically encountered in the coastal and offshore engineering field.

In chapter 6, the detailed approaches for parallelising the 3D version of numerical model developed in chapter 5 for fixed structures are given. A few test cases of wave interaction with a single cylinder and multiple cylinders are used to validate the 3D parallel model.

In chapter 7, conclusions are drawn and recommendations for future work are made.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a review of some numerical techniques used to model free-surface flows and fluid-structure interactions. These include not only general techniques such as those for solving the nonlinear transport terms in the NS equations, but also special issues inherent in PIC methods such as velocity interpolation between particles and grid.

2.1 Techniques for the advection terms

How a fluid field is advected differs in Eulerian and Lagrangian schemes, of which, the former refers to solving the advection term based on a fixed grid while the latter refers to those used in, for example, meshless methods such as SPH. For Lagrangian schemes, the fluid body is initially discretised into individual fluid elements such as particles. It is thus straightforward to advect the fluid elements by updating their positions. The primary advantages of these schemes are that the approaches employed to advect the fluid elements can be accurate to high-order while being computationally cheap at the same time (Smolianski et al., 2007). Moreover, the information stored on each individual element does not change while the elements are being advected, regardless of spatial distribution. Therefore, the Lagrangian advection schemes do not cause numerical diffusion to the fluid fields while they are being advected (Bridson, 2008). A typical example is that mass diffusion is eliminated in "classical" PIC by storing mass on the particles (Brackbill et al., 1988). In this sense, the Lagrangian schemes are well suited to handling convective transport of fluid properties.

By contrast, Eulerian schemes do not hold fluid properties using Lagrangian elements but the entire fluid fields on the grid. Complex calculations must be done in order to advect the fluid fields on the grid, both accurately and stably, particularly in regions where discontinuity occurs. Various schemes have been developed in the literature. While low-order methods (e.g. the first-order upwind differencing) are very diffusive when it comes to handling discontinuity, high-order approaches (e.g. the second-order centred differencing) can produce unstable oscillations (Griebel et al., 1998). There are nevertheless high-order schemes (e.g. the total variation diminishing (TVD) scheme) that are usually based on the idea of using a limiter that is applied in a way that smooth portions of a solution remain high-order accurate while sharp areas (e.g. discontinuous portions) remain nonoscillatory (LeVeque, 2002).

A third option is the coupled Eulerian and Lagrangian schemes such as those used in the Particle Transport Method (PTM) (Smolianski et al., 2007), the semi-Lagrangian method (Stam, 1999) and the "classical" PIC method (Harlow, 1964). These methods are similarly based on the concept of transporting the fluid fields in a Lagrangian manner while representing them on the grid. For example, the semi-Lagrangian method of Stam (1999) solves the advection of the fluid velocity at a grid node by backtracing this node through the velocity field on the grid for a time step to find its hypothetical "old" location and using the interpolated velocity at that location. This method is unconditionally stable as the velocity field is interpolated from the previous one and is thus bounded. However, because interpolation is used every time step the induced numerical diffusion could be large. Different to the use of hypothetical nodes, the PTM (Smolianski et al., 2007) and "classical" PIC method (Harlow, 1964) use predefined particles to advect the fluid velocity field, but need to transfer the fluid velocity field between the particles and grid. By employing particles, both methods have the advantages of the Lagrangian advection scheme as mentioned above. However, the additional projection step is very likely to produce unwanted numerical diffusion. Therefore, high-order schemes are commonly required for the interpolation between the particles and grid; these are reviewed in the following section. For other investigations that incorporate the coupled advection schemes, the reader is also referred to the works of Fedkiw et al. (2001), Ng et al. (2009) and Edwards and Bridson (2012).

2.2 Interpolation between grid and particles

As mentioned above, a double grid system is inherent in PIC methods. Therefore, information transfer between the two grid systems is necessary. In frameworks that incorporate a uniform grid and scattered particles, the grid-to-particle interpolation is straightforward and high-order accurate schemes (e.g. the fourth-order WENO scheme) can be easily applied. On the other hand, the particle-to-grid interpolation is somewhat complicated when particle distribution becomes irregular. In this section the particle-to-grid interpolation schemes that could be used in PIC methods are reviewed.

Scattered data interpolation/approximation is a very important and general issue in many aspects such as signal processing, statics and herein particle-based methods in CFD modelling. Particularly, this issue is well studied by the SPH community, where the particle approximation using a kernel function (see Equation 1.1) is fundamental yet a large source of truncation error. Therefore, attention has been focused on the various kernel interpolation methods used in the SPH community. First of all, Monaghan and Lattanzio (1985) proposed probably the most commonly used cubic spline kernel for SPH simulations. However, Liu et al. (2003) pointed out that the kernel interpolation cannot be formally high-order at the particle approximation, when the particle distribution is irregular, which is usually the case in the modelling of hydrodynamic problems. They instead proposed a general approach of constructing a kernel function of desired order of accuracy via considering the particle distribution. However, the constructed kernel function maybe partially negative, not symmetric and not monotonically decreasing, and as such, it could lead to some severe consequences such as breakdown of computations (Liu and Liu, 2003). Because of this, approaches that do not change the conventional kernel functions are usually more preferable, as they are more stable. For example, Chen and Beraun (2000) developed a generalised SPH model by normalising the kernel interpolation, which achieves at least a first-order accuracy even around boundary regions where the kernel interpolation is truncated. Liu and Liu (2006) further developed this approach to second-order accuracy, at the cost of having to solve a system of equations at each particle position. This is apparently very costly when a large number of particles are employed, not to mention the potential instability involved in the solution of the system of equations when it comes to strong dynamic problems.

Another notable type of methods for scattered data interpolation are those from the least square (LS) family. In particular, the Moving LS (MLS) method holds the idea of constructing functions that "best fit" the values at scattered points within a support area. Through choosing different basis of functions, the MLS method can achieve arbitrary order of accuracy. For example, Edwards and Bridson (2012) adopted the MLS approach for the particle-to-grid interpolation in their high order PIC scheme which achieves a fourth-order convergence on a variety of problems. It is nevertheless noteworthy that no applications to free-surface flows are made in their work. Dilts (1999, 2000) proposed using the MLS interpolation as a substitution of the kernel interpolation in the SPH method. However, similar to some aforementioned kernel interpolation methods, the MLS interpolation generally requires solving a system of equations at each particle position. Similarly, the coefficient matrix of the system of equations can become singular when the number of neighbouring particles is less than the order of the basis of functions (Dilts, 1999), which could easily happen in regions around the free surface when modelling free-surface flows.

2.3 Free-surface capturing and tracking

An essential capability of the numerical model developed in this thesis is the handling of moving free surfaces of fluid flows. The numerical techniques available in the literature for this are mainly categorised into two types, depending on whether the computational grid is movable or fixed (Jafari and Ashgriz, 2013). In the first category (herein refer to as the moving grid method), the grid points are moved according to the flow characteristics to naturally track the free surface. In cases where all grid points are movable, the method is Lagrangian (e.g. the SPH method). In the second category (herein refer to as the fixed grid method), the methods employ an additional medium to track the free surface in lieu of moving the grid. The fixed grid method can be further divided into equation-based and particle-based methods according to the medium selected. Examples of equation-based methods include the VOF method (Hirt and Nichols, 1981) and Level Set method (Osher and Sethian, 1988). For the particle-based methods, usually a set of marker particles either at the interface or in the entire fluid domain are employed to track the free surface akin to the moving grid methods. In the following, typical methods in the category of fixed grid methods are reviewed, as they are more close to the hybrid nature of the PIC

methods to be studied in this thesis.

Particle-based method

One of the earliest free-surface capturing methods is the marker-particle technique that is used in the PIC method and the MAC method. In the original MAC method (Harlow and Welch, 1965), the marker particles are initially seeded in the entire fluid region and then used to track the fluid configuration according to the fluid velocity on the grid. In particular, cells having no particles are regarded as air cells, and correspondingly, those containing particles are defined as water cells. This technique can easily handle complex interfaces with large deformations due to its particle nature. On the other hand, to achieve a well-defined interface, this approach requires far more marker particles than the computational grid cells (Hyman, 1984), which can be prohibitively expensive. In contrast, another type of particle-based methods, proposed in, for example, Unverdi and Tryggvason (1992) and Tryggvason et al. (2001), resolve these issues by seeding, linking and tracking the particles only at the interface. The (surface) marker particles move with the interface and thus naturally track its evolution through the simulation. An obvious advantage of these methods is that they are more accurate than the MAC method, as the free surface is represented at a sub-grid scale. Furthermore, computations of the characteristics of the free surface such as curvature are straightforward. The disadvantage, however, is their lack of abilities in dealing with free-surface folding, merging or breaking, particularly in 3D simulations, where the complexity in organising marker particles at the surface increases significantly that may even fail when using this approach.

VOF method

The VOF method proposed in Hirt and Nichols (1981) became very popular quickly after its invention, as it offers a simple and economical way to capture the free surface. The main idea is that, in each computational cell, a volume fraction, F, is stored and defined in a manner that F equals 1 in cells that are fully occupied by a fluid, and F equals 0 in cells without such fluid. A value between 0 and 1 represents that the cell is partially filled with the fluid. In this sense, the VOF method is very efficient in terms of computational resources, especially for 3D calculations. On the other hand, as suggested in Hirt and Nichols (1981), updating the volume value F through the simulation is relatively complicated. In Hirt and Nichols (1981), the

time dependent governing equation of F is given as:

$$\frac{\partial F}{\partial t} + \boldsymbol{u} \cdot \nabla F = 0\,. \tag{2.1}$$

where $\boldsymbol{u}$ is the velocity defined on the grid. Equation 2.1 ensures that the change of F in a cell equals the overall flux through the cell faces. To solve this equation, Hirt and Nichols (1981) used a donor-acceptor flux approximation technique. In the same work, the free surface is reconstructed based on F as "steps" that align with the grid axis. This is similar to an earlier work in Noh and Woodward (1976), where the use of a straight line that is either parallel to or perpendicular to the grid axis (one for each axis) is introduced. This method is referred to therein as the SLIC (simple line interface calculation) method and is suitable for multiphase flow simulation due to its simple line assumption. However, the SLIC method is low-order accurate and generates a large amount of flotsam (Scardovelli and Zaleski, 1999). Later, Youngs (1982) developed the PLIC (piecewise linear interface calculation) approach, where a piecewise linear interface geometry is constructed within each cell, considering all volume values in the surrounding cells. It is noteworthy that in the past few decades the VOF method has been widely used and developed based on the above-mentioned methods.

Level Set method

The Level Set method was devised by Osher and Sethian (1988) and has been successfully applied by the CFD community as an efficient free-surface capturing technique (see, e.g., Sussman et al. (1994)). Different from the VOF method, the idea here is that a function $\phi(x,t)$ is defined as the shortest distance from a single point to a specific surface, which is the contour line of $\phi(x,t) = 0$. The level-set function $\phi(x,t)$ can have opposite signs for the domains in both sides of the surface and in this case it is called "signed distance function". For example, for water-air flow problems, if $\phi(x,t) < 0$ denotes the water region, then $\phi(x,t) > 0$ indicates the air region. The free surface is thus captured through finding the zero contour during the numerical modelling. Similar to the VOF method, the evolution of the level-set function $\phi(x,t)$ is also resolved via a transport equation:

$$\frac{\partial \phi(x,t)}{\partial t} + \boldsymbol{u} \cdot \nabla \phi(x,t) = 0\,. \tag{2.2}$$

where $\boldsymbol{u}$ is the velocity defined on the grid. The spatial derivative of Equation 2.2

is usually solved by high-order finite difference schemes such as the essentially non-oscillatory (ENO) scheme and Weighted ENO scheme (Jiang and Peng, 2000), as low-order schemes could upset the mass conservation law, especially in under-resolved regions (Rider and Kothe, 1995). Sussman and Puckett (2000) coupled the Level Set method with a high-order VOF method. In their approach the mass is well conserved by the VOF method and characteristics of the free surface such as the mean curvature are straightforwardly derived from the Level Set method. Foster and Fedkiw (2001) proposed the use of particles seeded at one side of the zero contour to improve the local accuracy of the Level Set method. Later, Enright et al. (2002) improved this idea by using particles at both sides of the zero contour. They claimed that the use of particles allows the Level Set method to obtain a sub-grid scale accuracy around the interface, and to counteract the detrimental mass loss of the Level Set method in under-resolved regions.

2.4 Solution algorithms for the Navier-Stokes equations

In this section, the solution algorithms for the incompressible NS equations are reviewed. It is known that the main difficulty in getting a time-dependent solution for the incompressible NS equations lies in the fact that the continuity equation does not explicitly contain the time derivative or the pressure information (Kim and Moin, 1985); the pressure and velocity are coupled with one another in the equation system. This is different from the solution algorithms for the compressible NS equations, where the pressure is explicitly calculated via a state equation (see e.g. the Weakly compressible SPH (WCSPH) solver in Colagrossi and Landrini (2003)).

The algorithms for solving the incompressible NS equations generally, as pointed out in McDonough (2013), fall into two categories. The first employs the pressure as a Lagrange Multiplier to satisfy the mass conservation constraint resulting in having to solve a pressure Possion equation (PPE). Examples of this category include the MAC method (Harlow and Welch, 1965), the pressure projection method (Chorin, 1968) and the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) technique (Patankar and Spalding, 1972), as well as their variants. The other method, in contrast, directly modifies and solves the continuity equation for pressure, such as the artificial compressibility method (Chorin, 1967; Kwak et al., 1986). Here attention

has been focused on the algorithms in the first category, particularly the pressure projection method and the SIMPLE-type methods as mentioned above, as they have been widely used in modern CFD solvers such as OpenFOAM® (The OpenFOAM® Foundation, 2014), which is one of the most widely used CFD tools currently in the coastal and offshore engineering community (see e.g. Higuera et al. (2013b)).

The pressure projection technique was first proposed by Chorin (1968, 1969) to decouple the pressure and velocity in the incompressible NS equations. The idea being that, based on the Helmholtz-Hodge decomposition, the pressure is employed as an operator to project any divergent velocity field onto its divergence-free part. Typically, an intermediate velocity field is first obtained by solving the convection-diffusion part, as well as source terms, of the momentum equation. Then, a PPE is constructed by combining the continuity equation with the remaining pressure gradient part of the momentum equation. Finally, after the pressure has been solved for, it is used to project the intermediate velocity onto its divergence-free part. In this solution procedure, the selection of boundary conditions and numerical schemes for the intermediate velocity plays an important role in the accuracy of both the velocity and the pressure (Brown et al., 2001). The original scheme of Chorin (1968) is only first-order accurate in time. Subsequently, extensions have been made to improve the accuracy. For example, second-order accurate pressure projection schemes can be found in Kim and Moin (1985); Van Kan (1986); Choi and Moin (1994) and many others. These methods usually involve an implicit treatment of the convection and diffusion terms, requiring well-posted Dirichlet boundary conditions for the velocity and Neumann conditions for the pressure. It is nevertheless noted that free-surface problems were *not* considered by these papers. In fact, very few high-order projection methods were developed for modelling free-surface flows (see, e.g., Sussman (2003) and Yang and Prosperetti (2006)). One reason for this, as suggested in Yang and Prosperetti (2006), is that the presence of free surface requires modifications on the associated boundary conditions and hence prevents a direct application of the earlier approaches. The existing high-order projection schemes for free-surface flows are somewhat sophisticated and limited to certain problems. Since the focus of this thesis is not on developing a high-order PIC scheme, no further details regarding those schemes are reviewed herein.

Another well-known scheme for solving the NS equations is the SIMPLE scheme, which was first developed by Patankar and Spalding (1972). The core of this so-

lution scheme is the introduction of a pressure correction term. The scheme first solves the momentum equation for an intermediate velocity field based on the initial values of the pressure and velocity. Then, a pressure correction term is utilised, for which a pressure correction equation (basically a Poisson equation) that enforces the continuity equation is formed and solved. Finally, the pressure correction term is used to update the pressure and velocity. The whole procedure is run iteratively until convergence. The convergence rate of the SIMPLE scheme is relatively slow (Patankar, 1980). In order to improve the efficiency, the SIMPLE-Revised method was developed by Patankar (1980). The two methods differ in the computation of the pressure field. In the SIMPLE-Revised method, the pressure is obtained by solving a PPE that is formed similar to the aforementioned pressure projection method, and this pressure is then used as an initial guess for the calculation of the pressure correction term following the SIMPLE approach. However, it is noted that, in the SIMPLE-Revised method, the pressure correction term is only used to update the velocity. The whole procedure is again repeated until convergence, which, to some extent, still limits the efficiency of such scheme despite the improvement. Therefore, an important extension of the SIMPLE-type scheme is the non-iterative pressure implicit with splitting of operator (PISO) algorithm (Issa, 1986), which was originally devised for solving time-dependent flow problems. In general, when applied to solving incompressible flows, the PISO scheme incorporates at least two correction steps, each of which is akin to that of the aforementioned pressure projection method where the pressure and velocity from the previous step are regarded as guesses. Since iteration is avoided, the PISO algorithm is shown to be much more efficient by Issa et al. (1986). Nevertheless, it is noteworthy that both the SIMPLE-type and PISO method have been widely used in modern CFD solvers such as the OpenFOAM® model.

Generally, a Poisson equation, i.e. a linear system of equations, needs to be resolved during the solution of the incompressible NS equations. There are many numerical approaches currently available for solving this kind of system of equations (see e.g. Press et al. (1992)). These include both direct solution and iterative solution. Typical examples of direct solvers are the Cramer's rule, Gauss elimination and LU decomposition. For iterative solvers, there are traditional Jacobi, Gauss-Seidel methods as well as more efficient Conjugate Gradient (CG) method for linear positive definite and symmetric system. For a comprehensive analysis of solving the system of equations, the reader is referred to Saad (2003).

2.5 Algorithms for fluid-structure interaction

A main feature of the PIC model to be developed is the capacity to model fluid-structure interactions, in particular, wave-structure interactions, in the coastal and offshore engineering field. The focus of this section is on the fluid-structure coupling approaches that are suitable for numerical solvers with a fixed Eulerian grid, as the main calculations of PIC methods are based on the grid. Methods that are computationally feasible for both fixed and movable structures are of principal interest.

Peskin (1972) first devised the Immersed Boundary (IB) technique for fluid-structure interaction. The main idea of this approach is that, with a mixture of Eulerian and Lagrangian variables, the structure boundary is represented by discrete Lagrangian markers embedding in and exerting forces to the Eulerian fluid domain (Su et al., 2007). The interaction between the Lagrangian markers and the Eulerian grid is conducted using a discretised delta function. The boundary forces exerted on the fluid domain is computed based on the configuration and the diversity (e.g. elasticity) of the structure boundary. For this reason, the IB approach has certain advantages in simulating the complex interaction of fluid with moving elastic structures, such as flow through the natural mitral heart valve as shown in Peskin (1972). Meanwhile, the IB approach has also been applied to simulate fixed and rigid structures; see e.g. Lai and Peskin (2000), Mohd-Yusof (1997) and Su et al. (2007). In these papers, the technique used to maintain structure fixedness and calculate the boundary forces differs. For example, Lai and Peskin (2000) linked the Lagrangian markers of the structure boundary to their fixed equilibrium positions using a stiff spring function and calculated the boundary forces simply using this function when the boundary markers fall away from the desired location. The periodic nature of the spring function, nevertheless, can lead to errors such as force oscillations in fluid regions near the structure boundary. Differently, Mohd-Yusof (1997) and Su et al. (2007) calculated the boundary forces based on the concept that the fluid velocity should be forced to the prescribed velocity on the solid boundary (zero for a fixed structure). This method, as demonstrated in Su et al. (2007), is also capable of simulating moving structures with prescribed motions within the IB framework.

Batty et al. (2007) proposed a novel variational framework approach that is based on the pressure projection method. The feature of their method is that the pressure

field of fluid is reinterpreted as a Lagrange multiplier that minimizes the kinetic energy of the whole system, including both structures and fluids. This means, in the presence of rigid structures, the fluid pressure is implicitly solved for by setting the derivative of the kinetic energy of the whole system, with respect to the fluid pressure, to zero. This idea is analogous to the treatment of inelastic contact force between rigid bodies. Furthermore, the major computation of this approach is on the fixed Eulerian grid; structures are represented by a set of volumes of the grid cells, which is very similar to the cut cell approach discussed as follows.

Despite its novelty in handling fluid-structure interaction, the variational framework approach of Batty et al. (2007) is not convergent in the L^∞ norm for problems involving fixed structures (Ng et al., 2009), and thus this approach is not recommended for computations where the velocity field around objects is important. Instead, Ng et al. (2009) proposed a Cartesian cut cell method, which was shown to be convergent and accurate. An essential part involved in this cut cell approach is the computation of an integral of velocity fluxes along the structure boundary, which, however, appears to be non-trivial for moving boundaries that are non-grid aligned. Also, Su et al. (2007) suggested that the complex cut cell shape can unavoidably increase CPU cost, due to the necessity of a sophisticated interpolation procedure for approximating the fluxes in the irregular-shaped cut cells. Nevertheless, the Cartesian cut cell approach has been widely used as an alternative to unstructured grid techniques for structures with complex body shapes (Noh, 1963; Purvis and Burkhalter, 1979; Quirk, 1994; Qian et al., 2006). It offers great advantages when moving boundaries are involved, as there is no need to re-mesh the computational domain. In the context of inviscid compressible flow, Noh (1963) first suggested combining the idea of a cut cell fraction with an application of the finite volume method (FVM) to treat (deformable) solid boundaries. A similar approach was also proposed by Purvis and Burkhalter (1979) to solve the equations of transonic potential flow. This procedure was then further investigated by Ng et al. (2009) with regard to the order of convergence and accuracy; their results show second-order accuracy in both the L^1 and L^∞ norms in two spatial dimensions.

Another technique for handling fluid-structure interaction, particularly particulate flows, is the Distributed Lagrange Multiplier (DLM) method proposed by Glowinski et al. (1999) and Patankar et al. (2000). This approach is capable of handling the interaction of fluids with a large number of structures. The novelty of this approach,

as noted by Carlson et al. (2004), is that the structures are first resolved on the grid exactly as if they were fluids. Then, corrections are made to the grid velocity within the structure phase, accounting for the density differences between structures and fluids. Finally, if rigid structures are simulated, the rigidity constraint is enforced to obtain the final unique structure velocity. The enforcement of the rigidity constraint, as noted in Patankar et al. (2000), produces an additional stress tensor within the structure phase that is akin to the pressure as a Lagrange multiplier for enforcing fluid incompressibility. It is noteworthy that, later, Patankar (2001) proposed a fast and implicit solution of the rigidity constraint that conserves the mass and momentum. Furthermore, while Patankar (2001) used a finite-element formulation for this approach, Carlson et al. (2004) applied it within a finite difference framework. Ardekani et al. (2007) also modified this approach by separately solving for the components of the structure velocity on a staggered grid, in order to achieve a higher accuracy for the structure velocity.

2.6 Summary

In this chapter techniques for the core components required in CFD modelling were reviewed. Particular attention was focused on techniques that are related to the use of Lagrangian particles and an Eulerian grid. For each aspect, detailed approaches differ in the environments they are developed and tend to have both advantages and disadvantages. For solving the advection terms and tracking free surfaces, it is found that while grid-based methods require certain complexities to achieve a high-order accuracy, particle-based methods are more straightforward. Information transfer between scattered particles and uniform grid, as an important component inherent in PIC methods, is also discussed. It is noted that lower-order methods are more stable at the cost of accuracy, while high-order methods are usually unstable especially for problems involving strong hydrodynamic phenomena for which significant particle disordering tends to occur. Furthermore, various techniques for fluid-structure interactions in grid-based solvers are reviewed; these help provide solutions of developing fluid-structure interaction schemes that work in the PIC framework.

CHAPTER 3

A FULL PARTICLE PIC SOLVER FOR FREE-SURFACE FLOWS

This chapter describes the methodology of the proposed "full particle" PIC solver for free-surface flows, which is inspired by the work presented in Kelly (2012). The fluid-structure interactions are not included here but in the following chapters. Without loss of generality, the numerical methods in this chapter are presented in two spatial dimensions, unless otherwise stated.

3.1 Overall scheme of the numerical solver

3.1.1 Governing equations

The model solves the incompressible Newtonian NS equations for single-phase fluid flows:

$$\nabla \cdot \boldsymbol{u} \;=\; 0, \tag{3.1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \boldsymbol{f} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u}, \tag{3.2}$$

where, in two spatial dimensions, $\boldsymbol{u} = [u, w]^T$ is the velocity field; t is the time; p is the pressure; $\boldsymbol{f} = [0.0, -9.81]^T$ represents the body force due to gravity; ρ is the liquid density and ν is the kinematic viscosity of the fluid. Equation 3.1 is the continuity equation that preserves the mass conservation, and Equation 3.2 is the momentum equation.

According to the PIC methodology, both Eulerian grid and Lagrangian particles are employed to solve the NS equations. Particularly, in the present "full particle" PIC model, the particles carry the mass and momentum of the fluid, while the underlying grid is solely used for computational convenience. The solution algorithm here is divided into two major stages: an Eulerian stage and a Lagrangian stage. In the Eulerian stage, the NS equations, ignoring the nonlinear advection term (the second term at the left hand side (LHS) of Equation 3.2), are first solved on the underlying grid in an Eulerian sense. Following that, in the Lagrangian stage, the remaining advection term is solved using the particles in a Lagrangian manner. It should be noted that, in the above solution procedure, the particles are used to track the fluid configuration, and the velocity field carried by the particles is first mapped to the grid prior to the Eulerian stage calculation and then updated in the Lagrangian stage calculation. The detailed numerical methods involved in each stage are described in the following sections.

Figure 3-1 gives an overview of the computational domain Ω, as well as its discretisation by the grid and particles. The uniform staggered grid proposed by Harlow and Welch (1965) is employed in the present model. By using the staggered grid, the pressure values are stored at cell centres that are, for example, denoted by (i, j), and the velocity values are recorded at the centres of relevant cell edges, which are represented by, for example, $(i + \frac{1}{2}, j)$. Also, following Harlow and Welch (1965), a layer of cells beyond the domain boundary, $\partial\Omega$, are used to enforce the boundary conditions, and these cells are hereafter referred to as the ghost cells. For solving the single-phase free-surface water flow as sketched in Figure 3-1, grid cells occupied by the particles are marked as the water cells, while those having no particles inside are marked as empty cells. The free-surface position is tracked by the particles. To achieve a good resolution, more particles than grid cells are commonly used. The number of particles, however, is clearly a trade-off between accuracy and cost. Obviously, a large number of particles can give a better representation of fluid body, and thus facilitate the treatment of free-surface boundary conditions. On the other hand, the use of more particles requires more memory storage, which is already very demanding in PIC due to the double-grid system. For the current "full particle" PIC model, initial configurations of four particles per cell in two spatial dimensions and eight particles per cell in three spatial dimensions are utilised; both have proven to be satisfactory. For a discussion of the initial number of particles to use in "classical" PIC, however, the reader is referred to Harlow (1964).

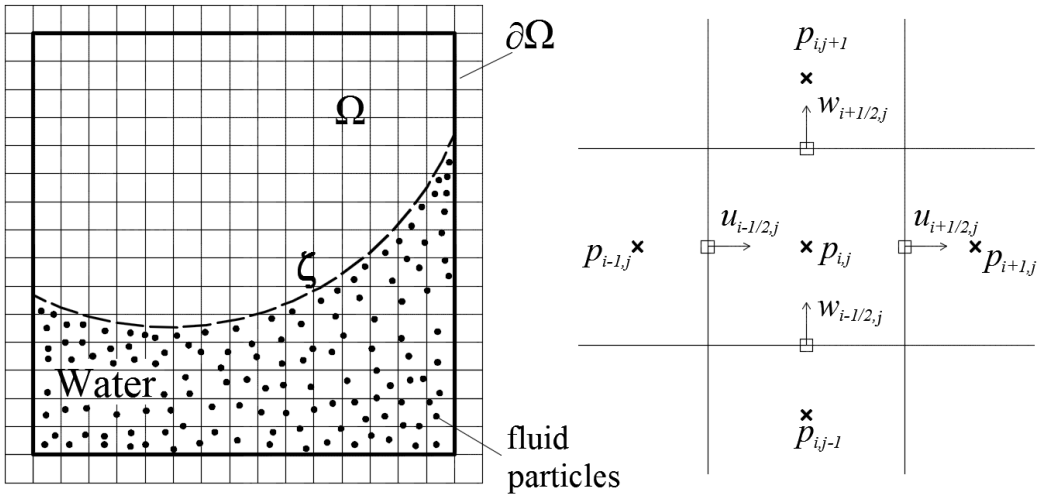


Figure 3-1: The discretisation of the computational domain by particles and grid (left panel). The right panel presents the staggered grid where pressure is stored at cell centres (cross) and velocity components are recorded at the centres (white square) of the vertical and horizontal cell edges, respectively.

3.1.2 Eulerian stage

In this stage, the NS equations, ignoring the nonlinear advection term (the second term at the LHS of Equation 3.2), are solved on the grid. Prior to the solutions, the velocity field $\boldsymbol{u}^n$ at the nth time-step on the grid is mapped from the velocity field carried by the particles; this is discussed in Section 3.2.2.

The pressure and velocity are decoupled and solved for, using the pressure projection method of Chorin (1968). A tentative velocity, $\tilde{\boldsymbol{u}}$, is first given by applying the body force and viscosity term using the first-order accurate explicit Euler method in time:

$$\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^n}{\Delta t} = \nu \nabla^2 \boldsymbol{u}^n + \boldsymbol{f}, \tag{3.3}$$

where Δt is the time step. The first step is to apply the body force:

$$\boldsymbol{u}^1 = \boldsymbol{u}^n + \Delta t \boldsymbol{f} \ . \tag{3.4}$$

The viscosity term is then resolved using a simple forward-in-time centered-in-space (FTCS) difference scheme. For example, the velocity component in x-direction at

point $(i+1/2, j)$ is calculated as:

$$\tilde{u}_{i+\frac{1}{2},j} = u^1_{i+\frac{1}{2},j} + \nu\Delta t\left(\frac{u^1_{i+\frac{3}{2},j} + u^1_{i-\frac{1}{2},j} - 2u^1_{i+\frac{1}{2},j}}{(\Delta x)^2} + \frac{u^1_{i+\frac{1}{2},j+1} + u^1_{i+\frac{1}{2},j-1} - 2u^1_{i+\frac{1}{2},j}}{(\Delta z)^2}\right), \tag{3.5}$$

where Δx and Δz are the cell lengths in x-direction and z-direction, respectively; a uniform grid is utilised in the present model, i.e. $\Delta x = \Delta z$. The velocity components in other directions are computed in the same manner. It is noted that as the viscosity term is resolved explicitly in time, the time step must be restricted by this term in order to ensure numerical stability (Morris et al., 1997). This constraint can become dominant when the fluid viscosity is high and/or the grid size is very fine. The calculation of the time step is detailed in Section 3.2.5.

After resolving the body force and viscosity term, the tentative velocity $\tilde{\boldsymbol{u}}$ is unlikely to be divergence-free, and the next step is to find a pressure field p^{n+1} to maintain the incompressibility condition (Equation 3.1). This is achieved by solving a pressure Poisson equation (PPE). Recalling the remaining part of the momentum equation, we have:

$$\frac{(\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}})}{\Delta t} = -\rho^{-1}\nabla p^{n+1}. \tag{3.6}$$

Taking the divergence of both sides of the above equation and recalling that the new velocity field, $\boldsymbol{u}^{n+1}$, must be divergence-free, gives a PPE for the pressure:

$$\Delta t\rho^{-1}\nabla^2 p^{n+1} = \nabla\cdot\tilde{\boldsymbol{u}}\,. \tag{3.7}$$

This PPE are discretised and solved using a finite difference scheme that includes the treatment of both domain and free-surface boundary conditions. This is detailed in Section 3.2.1.

After solving the PPE, the pressure field p^{n+1} is found and the next step is to project the tentative velocity field, $\tilde{\boldsymbol{u}}$, onto a divergence-free velocity field, using the pressure as a Lagrange multiplier:

$$\boldsymbol{u}^{n+1} = \tilde{\boldsymbol{u}} - \Delta t\rho^{-1}\nabla p^{n+1}. \tag{3.8}$$

3.1.3 Lagrangian stage

In this stage, the remaining advection term of the momentum equation (the second term at the LHS of Equation 3.2) is solved via the particles in a Lagrangian manner. Once the divergence-free velocity field on the grid is obtained, the velocity itself or the velocity change, $\boldsymbol{a}^{n+1} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}$, on the grid is interpolated onto the particles to update the velocity they carry. Ideally, for the current "full particle" PIC solver, the velocity change should be used to increment the particle velocity, following the "full particle" PIC based FLIP code of Brackbill and Ruppel (1986). Since the interpolated velocity change, and the associated numerical error, are relatively small at each time step, the numerical dissipation occurring in this procedure is much smaller than that induced by directly interpolating the velocity itself as the "classical" PIC does. However, despite being small, the numerical errors associated with using the velocity change, are allowed to accumulate, rather than being filtered through the grid when directly interpolating the velocity itself (Jiang et al., 2015). This can lead to numerical instability issues. In the current model, this problem is alleviated by using the method proposed in Zhu and Bridson (2005). That is, a weight average between the two methods is used for the final particle velocity:

$$\boldsymbol{v}_p = c\boldsymbol{v}_F + (1-c)\boldsymbol{v}_C, \tag{3.9}$$

where $\boldsymbol{v}_p$ is the particle velocity; c is an empirical blending coefficient; $\boldsymbol{v}_F$ denotes the particle velocity incremented by the interpolated velocity change:

$$\boldsymbol{v}_F^{n+1} = \boldsymbol{v}_p^{n} + \sum_i \boldsymbol{a}^{n+1} S_i \ , \tag{3.10}$$

and $\boldsymbol{v}_C$ denotes the particle velocity directly replaced by the interpolated velocity itself:

$$\boldsymbol{v}_C^{n+1} = \sum_i \boldsymbol{u}^{n+1} S_i \ , \tag{3.11}$$

where S_i in the above two equations is the interpolation function discussed in Section 3.2.2. The chosen value of c should fit the purpose of stabilising the code while introducing as little unwanted numerical dissipation as possible. In this thesis, c is chosen to be 0.96. In practice, the change of this coefficient was found to have negligible influence when it is close to, but less than, unity. This is demonstrated in Section 3.3.

The particles, carrying the newly updated velocity field, are then moved to advect the fluid body. Since each particle carries an individual velocity, the multistreaming phenomenon may occur if the particles are moved using the velocity they carry (Brackbill and Ruppel, 1986). To eliminate this effect, following Brackbill and Ruppel (1986), the particles are moved through the divergence-free velocity field on the grid. In other words, the particles are advected using the velocity directly interpolated from the grid, $\boldsymbol{v}_C$, rather than particle velocity $\boldsymbol{v}_p$:

$$\frac{d\boldsymbol{x}_p}{dt} \;=\; \boldsymbol{v}_C, \tag{3.12}$$

where $\boldsymbol{x}_p$ is the particle position. Once the particles are advected, one computation cycle is completed.

Assuming $c = 1.0$ in Equation 3.9, the difference of using $\boldsymbol{v}_C$ instead of $\boldsymbol{v}_p$ for particle advection can be revealed by comparing the two velocities. Recalling Equation 3.10 and Equation 3.11, we have:

$$\boldsymbol{v}_C^{n+1} = \sum_i \boldsymbol{u}^{n+1} S_i = \sum_i (\boldsymbol{u}^n + \boldsymbol{a}^{n+1}) S_i = \sum_i \boldsymbol{u}^n S_i + \sum_i \boldsymbol{a}^{n+1} S_i \;; \tag{3.13}$$

$$\boldsymbol{v}_p^{n+1} = \boldsymbol{v}_F^{n+1} = \boldsymbol{v}_p^n + \sum_i \boldsymbol{a}^{n+1} S_i \;. \tag{3.14}$$

It can be seen from the above equations that the difference is in fact equivalent to the difference between $\sum_i \boldsymbol{u}^n S_i$ and $\boldsymbol{v}_p^n$. It is noted that, as aforementioned, the velocity field $\boldsymbol{u}^n$ on the grid is mapped from $\boldsymbol{v}_p^n$ after the particles have been advected in the last computation step and prior to the Eulerian stage solution in this time step. Therefore, $\sum_i \boldsymbol{u}^n S_i$ can be regarded as the velocity field $\boldsymbol{v}_p^n$ being filtered through the grid. The use of $\boldsymbol{v}_C$ thus may introduce an error for particle advection, whose order is determined by the interpolation schemes employed in the current numerical model. According to the discussion in Section 3.2.2, this could be formally first-order accurate. Nevertheless, the use of $\boldsymbol{v}_C$ to advect one particle considers the velocities carried by other particles around (via $\sum_i \boldsymbol{u}^n S_i$) and eliminates the multistreaming effect as the velocity is single-valued. It is noted that this idea, that the velocity from the momentum equation and the velocity used to change the positions need not be identical, is also well exploited for the SPH method (Monaghan, 1989).

As recommended by Bridson (2008), in the current numerical model, Equation 3.12 is integrated in time, using the third-order accurate Runge-Kutta scheme of Ralston

(1962):

$$y_{n+1} - y_n = \left(\frac{2}{9}\right) k_1 + \left(\frac{1}{3}\right) k_2 + \left(\frac{4}{9}\right) k_2 \tag{3.15}$$

where

$$\begin{aligned} k_1 &= \Delta t_n f(t_n, y_n) \\ k_2 &= \Delta t_n f(t_n + \frac{1}{2}\Delta t_n, y_n + \frac{1}{2}k_1) \\ k_3 &= \Delta t_n f(t_n + \frac{3}{4}\Delta t_n, y_n + \frac{3}{4}k_2) \end{aligned} \tag{3.16}$$

where Δt_n is the time step at n time-steps and $f(t_n, y_n)$ is the velocity interpolated to position y_n.

It is noteworthy that, before advecting the particles, the fluid velocity field must be extrapolated beyond the free surface, in order to correctly move the particles around the free surface. This extrapolation procedure is described in Section 3.2.3.

3.2 Numerical implementation

3.2.1 Solution of the PPE and treatment of boundary conditions

In this section, the approaches used to solve the PPE are described, together with the handling of the free-surface and domain boundary conditions.

Solution of the PPE

Equation 3.7 (the PPE) is discretised using a finite difference scheme as:

$$\begin{aligned} \frac{\Delta t(p^{n+1}_{i-1,j} - p^{n+1}_{i,j})}{\rho \Delta x} + \frac{\Delta t(p^{n+1}_{i+1,j} - p^{n+1}_{i,j})}{\rho \Delta x} + \frac{\Delta t(p^{n+1}_{i,j-1} - p^{n+1}_{i,j})}{\rho \Delta z} + \frac{\Delta t(p^{n+1}_{i,j+1} - p^{n+1}_{i,j})}{\rho \Delta z} \\ = \tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j} + \tilde{w}_{i,j+\frac{1}{2}} - \tilde{w}_{i,j-\frac{1}{2}} \ . \end{aligned} \tag{3.17}$$

By introducing a new variable $\tilde{\varphi} = \frac{\Delta t}{\rho \Delta x} p$, Equation 3.17 can be reorganised as:

$$4\tilde{\varphi}_{i,j} - \tilde{\varphi}_{i+1,j} - \tilde{\varphi}_{i-1,j} - \tilde{\varphi}_{i,j+1} - \tilde{\varphi}_{i,j-1} = - \left(\tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j} + \tilde{w}_{i,j+\frac{1}{2}} - \tilde{w}_{i,j-\frac{1}{2}} \right), \tag{3.18}$$

which forms a standard linear system of equations with a sparse coefficient matrix:

$$\mathbf{A}\tilde{\varphi} = \boldsymbol{b}\,, \tag{3.19}$$

where $\boldsymbol{b}$ is the source term and $\mathbf{A}$ is the coefficient matrix. The matrix $\mathbf{A}$ is also symmetric and positive definite. Thus, Equation 3.18 can be efficiently solved for p via many existing solvers, for which a review can be found in Saad (2003). In this thesis, the bi-conjugate gradient (BCG) method described in the numerical recipes of Press et al. (1992) is used. The calculation is terminated when the largest (in absolute value) component of the error, divided by the largest (in absolute value) component of $\tilde{\varphi}$, is less than a user defined number ϵ, which is set to 1.0×10^{-6} in the current solver. For finite time simulations, the use of a small ϵ ensures a low level of error in the calculation for pressure, and as such, the divergence-free condition is reasonably satisfied. Note that the solution in three spatial dimensions is handled in the same manner.

Treatment of the free-surface boundary condition

For water-wave type problems involved in this thesis, the kinematic viscosity of the fluid is relatively very small. Therefore, the dynamic boundary condition applied at the free surface is expressed as (Nichols and Hirt, 1971):

$$p = 0 \quad \text{on } \zeta(\boldsymbol{r}, t)\ . \tag{3.20}$$

where $\zeta = \zeta(\boldsymbol{r}, t)$ is the free surface and $\boldsymbol{r}$ is a position vector. Note that, as the particles are used to track the free surface, the kinematic condition is automatically satisfied by the particles during the Lagrangian stage described in Section 3.1.3.

The second-order accurate technique proposed in Gibou et al. (2002) is adopted to enforce the above boundary condition on the free surface defined by the particle positions. This is achieved via a signed distance function (SDF), ϕ, constructed at the centre of each grid cell, with respect to the free surface. For single-phase water-

wave problems, assuming $\phi < 0$ for regions within the water phase and $\phi > 0$ for regions on the other side of the free surface, $\phi = 0$ thus represents the free-surface iso-contour (iso-surface in three spatial dimensions). To construct such a SDF, the free surface represented by the particles is first reconstructed on the grid as initial values. The method proposed in Zhu and Bridson (2005) is employed for this purpose in the current model. That is, a set of initial values of the SDF at the closest empty cells (i.e. cells with no fluid particles) to the particle based free surface is computed by:

$$\phi_{ca}(\boldsymbol{X}_j) = |\boldsymbol{X}_j - \bar{\boldsymbol{X}}_0| - r_0 \tag{3.21a}$$

$$\bar{\boldsymbol{X}}_0 = \frac{\sum_i \boldsymbol{x}_i W_j(\boldsymbol{X}_j - \boldsymbol{x}_i, k_h)}{\sum_i W_j(\boldsymbol{X}_j - \boldsymbol{x}_i, k_h)} \tag{3.21b}$$

where ϕ_{ca} represents the initial value of the SDF; $\boldsymbol{x}_i$ and $\boldsymbol{X}_j$ denote the position vectors for a particle i and a grid node j, respectively; r_0 is a constant that may be interpreted as a particle radius; W_j is the cubic spline kernel function proposed in Monaghan and Lattanzio (1985), with the subscript j representing the index of the reference point $\boldsymbol{X}_j$; k_h is the so-called kernel length, which determines the neighbourhood of involved particles. Equation 3.21 can reconstruct the SDF of an isolated particle as well as flat or smooth surfaces but may have disadvantages for concave regions where $\bar{\boldsymbol{X}}_0$ may end up outside the fluid surface (Zhu and Bridson, 2005). In practice, the effects of this drawback are negligibly small when the resolution is high, as the neighbourhood of particles used are within just one or two cell lengths. Once the above procedure is completed, those initial SDF values, ϕ_{ca}, are then spread out onto a grid band around the free surface by solving the Eikonal equation:

$$|\nabla\phi(\boldsymbol{X})| = f(\boldsymbol{X}), \qquad \boldsymbol{X} \in \Omega \tag{3.22a}$$

$$\phi(\boldsymbol{X}) = \phi_{ca}(\boldsymbol{X}), \qquad \boldsymbol{X} \in \Gamma(\boldsymbol{X}, t) \in \Omega \tag{3.22b}$$

$$\phi(\boldsymbol{X}) = \phi_b(\boldsymbol{X}), \qquad \boldsymbol{X} \in \partial\Omega \tag{3.22c}$$

where $f(\boldsymbol{X})$ equals -1 inside the fluid domain and 1 outside the fluid domain; Ω and $\partial\Omega$ denote the computational domain and its boundary, where the boundary value, ϕ_b, is set to a very large number in the current implementation; $\Gamma(\boldsymbol{X}, t)$ represents the closest empty cells mentioned above, with the initial SDF value ϕ_{ca}. Equation 3.22 are solved by the fast sweeping approach proposed in Zhao (2004). It is noteworthy that, in the current model, the SDF is constructed at each time step using the above procedure, rather than being advected from an initial set up as the Level Set method

does (see, e.g., Enright et al. (2002)).

Once the SDF has been constructed, the free-surface boundary condition is applied using the approach proposed in Gibou et al. (2002). Because in the current model the pressure is stored and computed at the cell centres only for those with $\phi \leq 0$, the main idea is to linearly extrapolate an artificial pressure from the water cell ($\phi \leq 0$) to its adjacent cells with $\phi > 0$ such that the pressure at the free-surface position ($\phi = 0$) is zero. This, as demonstrated below, eventually leads to a rearrangement of the coefficient matrix, $\mathbf{A}$, of the PPE. Without loss of generality, this is explained in one spatial dimension for the solution of a PPE, $p_{,x,x} = \tilde{u}_{,x}$, where subscripts represent partial derivatives of the quantity. The free surface is located at x_f with $x_i < x_f < x_{i+1}$, where x_i and x_{i+1} denote the locations of the cell centres that are inside and outside the water phase, respectively. Figure 3-2 shows a sketch of this setup. The artificial pressure value in cell $(i+1)$ is thus expressed by:

$$p_{i+1} = (1 - \frac{1}{\Theta})p_i, \tag{3.23}$$

where $\Theta = \frac{x_f - x_i}{x_{i+1} - x_i}$. Note that in practice, for numerical stability the denominator Θ is modified to $\frac{x_f - x_i}{x_{i+1} - x_i} + \xi$, with ξ being a user-defined positive value, so that Equation 3.23 remains bounded and the computation is stable.

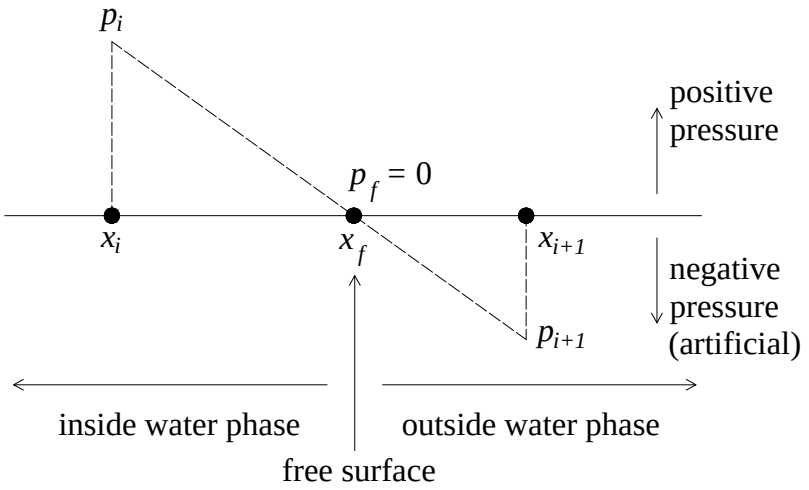


Figure 3-2: Sketch of the linear extrapolation for the artificial pressure.

Substituting Equation 3.23 into the discretisation form of the 1D PPE yields:

$$\frac{(\frac{0 - p_i}{\Theta \Delta x}) - (\frac{p_i - p_{i-1}}{\Delta x})}{\Delta x} = (\tilde{u}_{,x})_i \,. \tag{3.24}$$

From Equation 3.24 it is seen that the structure of the linear system of equations remains unchanged, and only the coefficients in front of the pressure on the LHS of Equation 3.24 have been modified to reflect the presence of the free surface.

For the 2D and 3D models in this thesis, the above modification is done dimension by dimension in the same manner, and Θ is calculated using the SDF value. Considering a 2D case and assuming that cell $(i+1, j)$ has been extrapolated an artificial pressure from cell (i, j), Θ is calculated by:

$$\Theta = \frac{|\phi_{i,j}|}{\phi_{i+1,j} + |\phi_{i,j}|} + \xi \,, \tag{3.25}$$

where $\phi_{i,j} \leq 0$ and $\phi_{i+1,j} > 0$ are the SDF values in cells (i, j) and $(i + 1, j)$, respectively, and ξ in the current model is set to $\frac{0.001\Delta x}{\phi_{i+1,j}+|\phi_{i,j}|}$. It is noted that the coefficient Θ is also used in the pressure projection step at the end of the Eulerian stage (see Equation 3.8). For example, in the above case, the velocity projection for $u_{i+1/2,j}$ is finally computed by:

$$\begin{aligned} u_{i+1/2,j} &= \tilde{u}_{i+1/2,j} - \frac{\Delta t}{\rho} \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \\ &= \tilde{u}_{i+1/2,j} - \frac{\Delta t}{\rho} \frac{(1 - \frac{1}{\Theta})p_{i,j} - p_{i,j}}{\Delta x} \end{aligned} \tag{3.26}$$

Treatment of the domain boundary condition

The domain boundary is grid-aligned as shown in Figure 3-1. The hybrid Neumann and Dirichlet conditions are applied at this boundary:

$$\boldsymbol{n} \cdot \boldsymbol{u}_b = 0 \text{ and } \boldsymbol{n} \cdot (\Delta t \rho^{-1} \nabla p) = \boldsymbol{n} \cdot (\tilde{\boldsymbol{u}} - \boldsymbol{u}_b) \text{ on } \partial\Omega, \tag{3.27}$$

where $\boldsymbol{u}_b$ is the grid-aligned boundary velocity. Following Griebel et al. (1998), the domain boundary conditions are also handled in a way that ultimately leads to modifications of the coefficient matrix of the PPE. For simplicity of presentation, the 1D standard PPE is again adopted as an example. The discretisation form of this equation is expressed as:

$$\frac{\Delta t}{\rho} \frac{p_{i+1} + p_{i-1} - 2p_i}{(\Delta x)^2} = \frac{\tilde{u}_{i+\frac{1}{2}} - \tilde{u}_{i-\frac{1}{2}}}{\Delta x}. \tag{3.28}$$

Assuming that cell $(i+1)$ is a ghost cell (i.e. cells beyond the domain boundary $\partial\Omega$, see Figure 3-1) and cell (i) is a fluid cell, so that the cell edge $(i+1/2)$ aligns with the domain boundary, where Equation 3.27 is applied:

$$u_{i+\frac{1}{2}} = \tilde{u}_{i+\frac{1}{2}} - \frac{\Delta t}{\rho}\frac{p_{i+1} - p_i}{\Delta x} = 0\,. \tag{3.29}$$

Substituting Equation 3.29 into Equation 3.28 leads to:

$$\frac{\Delta t}{\rho}\frac{0p_{i+1} + p_{i-1} - p_i}{(\Delta x)^2} = \frac{0 - \tilde{u}_{i-\frac{1}{2}}}{\Delta x}. \tag{3.30}$$

It can be seen from the above equation that, similar to the case of handling the free surface, only relevant coefficients of the PPE are modified to account for the grid-aligned domain boundary conditions. Note again that, in multiple dimensions, the same method is utilised in each direction. Furthermore, the free-slip condition is imposed on the domain boundary; this is done following Harlow and Welch (1965).

3.2.2 Interpolation between the particles and grid

During each time step, the velocity field, carried by the particles, is first mapped from the particles to the grid (see below, particle-to-grid interpolation) for the Eulerian stage computation (see Section 3.1.2), and then updated via the velocity field (as well as the velocity change) on the grid (see below, grid-to-particle interpolation) in the Lagrangian stage calculation (see Section 3.1.3). The interpolation schemes are discussed in this section.

Grid-to-particle interpolation

As a uniform staggered grid is utilised, the velocity interpolation (Equation 3.11), as well as the velocity change interpolation (Equation 3.10), from the grid to the particles is straightforward. One can easily adopt a bilinear (in two spatial dimensions) or trilinear (in three spatial dimensions) interpolation scheme as described in Press et al. (1992). However, in order to improve accuracy, the fourth-order WENO scheme proposed by Edwards and Bridson (2012) is employed in the current model. For simplicity of presentation, the implementation of a 1D case is shown here; extension to higher dimensions is componentwise. For four uniformly spaced points at

$$f = \frac{w1 * p1 + w2 * p2}{w1 + w2} \ ;$$

Function $(w1, p1) = F(f1, f2, f3, x)$;

Function $(w2, p2) = F(f2, f3, f4, 1 - x)$.

where

Function $(w, P) = F(f1, f2, f3, x)$

$d = (f3 - f1) * 0.5$;

$dd = f1 - 2 * f2 + f3$;

$S = d * (d + dd) + 4/3 * dd^2$;

$w = (2 - x)/(S + 1.\mathrm{E} - 06)^2$;

$P = f2 + x * (d + 0.5 * x * dd)$;

Figure 3-3: The 4th-order WENO algorithm proposed in Edwards and Bridson (2012).

(-1, 0, 1, 2) with values $(f1, f2, f3, f4)$, the algorithm used to calculate the value f at $x \in [0, 1]$ is listed in Figure 3-3. Note that in the vicinity of a domain boundary, $f1$ and $f2$ (or, $f3$ and $f4$), for example, may end up outside the domain boundary as shown in Figure 3-4. In this case, since a free-slip boundary condition is used, $f2$, located in the ghost cell, is set equal to $f3$ that is inside the computational domain. However, as only one layer of ghost cells are used in the current model, there is no real computational cell for $f1$. Thus, in this case, $f1$ is recorded in an imaginary cell and simply set to zero. This setting has proven to be satisfactory, for example, for the velocity interpolation around the domain boundary, by comparing the results with those obtained by the second-order conservative interpolation scheme of Meyer and Jenny (2004), where no additional stencils beyond the ghost cell are required. A comparison is presented in Section 3.3.

However, it should be noted that despite the fact that the fourth-order WENO scheme helps reduce the numerical dissipation due to truncation errors from the spatial discretisation, the overall accuracy of the current model cannot be expected to be formally high-order. This is due to the errors that arise from other numerical approximations; for instance, the particle-to-grid interpolation discussed below is formally first-order accurate.

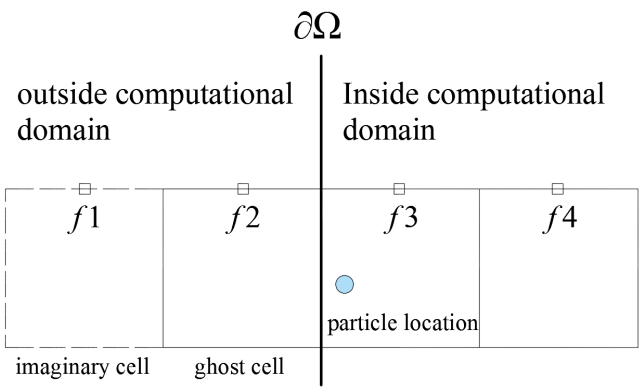


Figure 3-4: Schematic showing 4th-order WENO scheme for vertical velocity interpolation from grid to particle, in the vicinity of domain boundary.

Particle-to-grid interpolation

As mentioned in Section 3.1.2, prior to the Eulerian stage calculation, the velocity field carried by the scattered particles is mapped onto the grid. This mapping of the velocity field is somewhat involved, due to the inherent problem of PIC that the particle distribution tends to become uneven as the simulation progresses. The technique used here is based on the kernel interpolation, akin to the SPH method (Monaghan and Lattanzio, 1985). Both the mass and momentum carried by the particles are mapped to the grid in order to get the velocity field on the grid:

$$\boldsymbol{u}_g = \frac{\sum_{p\in\Omega_g} \boldsymbol{v}_p m_p \sigma_{pg}(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h)}{m_g} \ ; \tag{3.31}$$

$$m_g = \sum\nolimits_{p\in\Omega_g} m_p \sigma_{pg}(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h) \ , \tag{3.32}$$

where the subscript p and g represent indexes of a particle and a grid node (the centre of a cell edge in two spatial dimensions), respectively; $\boldsymbol{u}_g$ and m_g thus denote the fluid velocity and mass to be interpolated at the grid node g; $\boldsymbol{v}_p$ is the fluid velocity carried by the particle p; $\boldsymbol{x}_p$ and $\boldsymbol{X}_g$ are the position vectors for the particle p and the grid node g, respectively; σ_{pg} can be interpreted as a fraction of the mass, as well as the momentum, of the particle p that is assigned to the grid node g during this transfer, and it is computed by:

$$\sigma_{pg}(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h) = \frac{W_p(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h)}{\sum_{\tilde{g}\in\Omega_p} W_p(\boldsymbol{x}_p - \boldsymbol{X}_{\tilde{g}}, k_h)} \ ; \tag{3.33}$$

in the above equations, Ω_p and Ω_g denote circular areas (spherical areas in three spa-

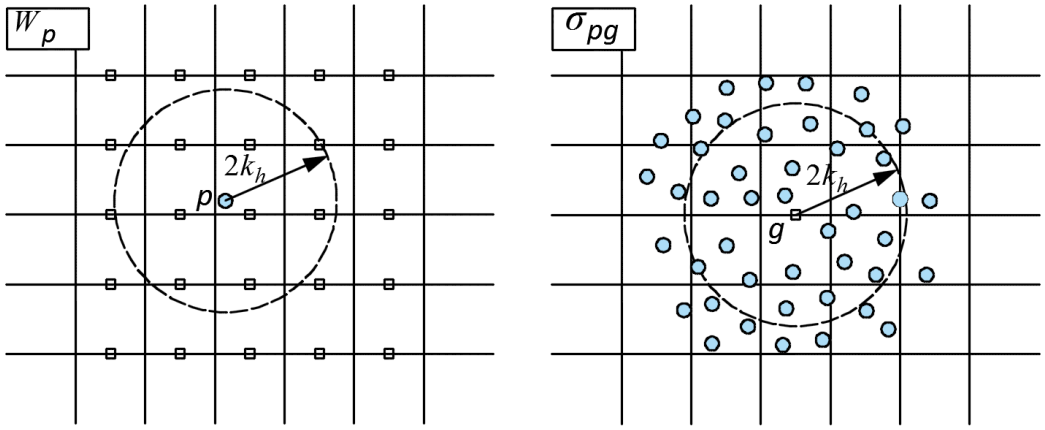


Figure 3-5: Schematic showing the kernel interpolation scheme for mass and momentum transfer from particles to grid: (a) compute σ_{pg}, using Equation 3.33; (b) compute particle-to-grid interpolation, using Equation 3.31 and Equation 3.32. The dots represent the particles, and the squares denote centres of cell edges.

tial dimensions) with particle p and grid node g located at the centres, respectively; k_h is the kernel length; W_p is the cubic spline kernel function proposed in Monaghan and Lattanzio (1985), with particle p being the reference point; $\tilde{g}$ represents the grid node whose velocity component is in the same direction as that of grid node g. It is noted that because a staggered grid is employed, Equation 3.31–Equation 3.33 are calculated in a dimension by dimension manner. Figure 3-5 shows a schematic of the mass and momentum transfer described in this section.

The cubic spline kernel function proposed in Monaghan and Lattanzio (1985) is expressed as:

$$W_i(\boldsymbol{r}_i - \boldsymbol{r}_j, k_h) = \alpha_D \begin{cases} \frac{2}{3} - S^2 + \frac{1}{2}S^3 & \text{if } 0 \leqslant S < 1, \\ \frac{1}{6}(2 - S^2) & \text{if } 1 \leqslant S < 2, \\ 0, & \text{if } S \geqslant 2, \end{cases} \tag{3.34}$$

where $S = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|/k_h$, and $\alpha_D = 15/7\pi k_h^2$ in two spatial dimensions and $3/2\pi k_h^3$ in three spatial dimensions, respectively.

It is worth noting that Equation 3.31 and Equation 3.32 conserve the mass and momentum during this mapping procedure, due to the fact that in any direction

$\sum_{g\in\Omega_p}\sigma_{pg}=1$. Taking the mass mapping (in one direction) as an example, this can be proved following Jiang et al. (2015):

$$\begin{aligned}\sum\nolimits_g m_g &= \sum\nolimits_g \sum\nolimits_{p\in\Omega_g} m_p\sigma_{pg} \\ &= \sum\nolimits_p m_p(\sum\nolimits_{g\in\Omega_p}\sigma_{pg}) \\ &= \sum\nolimits_p m_p \ .\end{aligned} \tag{3.35}$$

Note that, to satisfy the above equation, the areas Ω_p and Ω_g must be identical (by using the same radius) as depicted in Figure 3-5. This is due to the condition that a grid node that is found by a particle during the computation of σ_{pg} must also find this particle when summing up the fractions of mass and momentum received from the surrounding particles.

However, the interpolation through Equation 3.31 and Equation 3.32 cannot be expected to be formally high-order. This can be seen via the general form of the Taylor series expansion at the kernel approximation of a function $f(x)$ in one spatial dimension (Liu and Liu, 2006):

$$\begin{aligned}\int f(x')W(x-x',k_h)dx' =& f(x)\int W(x-x',k_h)dx' + \\ & f'(x)\int (x'-x)W(x-x',k_h)dx' + \\ & \frac{f''(x)}{2!}\int (x'-x)^2 W(x-x',k_h)dx' + ...\end{aligned} \tag{3.36}$$

Rewriting Equation 3.36 leads to:

$$\begin{aligned}\frac{\int f(x')W(x-x',k_h)dx'}{\int W(x-x',k_h)dx'} =& f(x) + \\ & f'(x)\frac{\int (x'-x)W(x-x',k_h)dx'}{\int W(x-x',k_h)dx'} + \\ & \frac{f''(x)}{2!}\frac{\int (x'-x)^2 W(x-x',k_h)dx'}{\int W(x-x',k_h)dx'} + ...\end{aligned} \tag{3.37}$$

Here, the second term at the right hand side (RHS) of Equation 3.37 is not necessarily zero when it is applied at the particle approximation. This is because when W is simply a symmetric kernel function such as the cubic spline kernel mentioned above, the irregular particle distribution makes the numerator nonzero. Therefore, Equa-

tion 3.37 is only first-order accurate for irregular particle interpolation. Nevertheless, Equation 3.37, as well as Equation 3.31, are stable and reproduce at least a constant field even around the boundaries where truncation exists. Thus, Equation 3.31 is used in the current model.

3.2.3 Velocity extrapolation into the atmospheric pressure region

For the current single-phase model, the divergence-free velocity field on the grid is restricted within the fluid region where SDF $\phi < 0$. However, in order to correctly conduct the grid-to-particle velocity interpolation using the 4th-order WENO scheme, and advect the particles around the free surface, the fluid velocity field on the grid may need to be extrapolated into the atmospheric pressure region where SDF $\phi \geq 0$. This section introduces the technique used in the current model for this velocity extrapolation procedure.

The Fast Marching Method proposed by Adalsteinsson and Sethian (1999) is employed to extrapolate the velocity by solving the following equation:

$$\nabla\phi \cdot \nabla q = 0\,, \tag{3.38}$$

where q is the velocity component on the grid, i.e. u and w. Equation 3.38 can be rewritten into:

$$\frac{\partial\phi}{\partial x}\frac{\partial q}{\partial x} + \frac{\partial\phi}{\partial z}\frac{\partial q}{\partial z} = 0\ . \tag{3.39}$$

Note that while the velocity is stored at cell edges, the SDF ϕ is recorded at cell centres. Hence, it is assumed that the SDF of a cell (i, j) and the velocity components in the positive directions, i.e. $u_{i+1/2,j}$ and $w_{i,j+1/2}$, correspond to each other when computing relevant gradients in Equation 3.39.

The velocity extrapolation procedure is conducted in a marching manner. The cell band that is closest to the free surface is first extrapolated with velocities. Then, these cells are marked up as if the free surface had marched in its normal direction and are used for the velocity extrapolation of further cell bands with respect to the free surface. This is repeated until the velocity field has been extrapolated to a certain number of bands of cells, which should be at least two for the current WENO interpolation.

To calculate the gradients in Equation 3.39 and hence compute the extrapolated velocity of a cell with $\phi \geq 0$, in each dimension the velocity and SDF values are taken from one of the two neighbouring cells, and this cell should be a fluid cell ($\phi < 0$) or belong to the previous band of cells where velocities have already been extrapolated. In the case that there are no such cells in one dimension, the velocity gradient in that dimension is set to zero. To further visualise the procedure, Figure 3-6 plots the configuration of a fluid body and the bands of cells where velocities are extrapolated in sequence. For instance, to extrapolate a velocity for the cell d in the first band as shown in Figure 3-6, the velocity and SDF values at cell e and b are used, as they are fluid cells with velocities. Therefore, let cell d be cell (i, j), then Equation 3.39 can be solved to get, for example, the vertical velocity $w_{i,j+1/2}$ at the edge $(i, j + 1/2)$ of cell (i, j):

$$\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \frac{w_{i+1,j+1/2} - w_{i,j+1/2}}{\Delta x} + \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta z} \frac{w_{i,j+1/2} - w_{i,j-1/2}}{\Delta z} = 0\,, \tag{3.40}$$

and since $\Delta x = \Delta z$, Equation 3.40 can be rearranged into:

$$w_{i,j+1/2} = \frac{(\phi_{i,j} - \phi_{i,j-1})w_{i,j-1/2} + (\phi_{i,j} - \phi_{i+1,j})w_{i+1,j+1/2}}{(\phi_{i,j} - \phi_{i,j-1}) + (\phi_{i,j} - \phi_{i+1,j})}\ . \tag{3.41}$$

Note that, normally, the horizontal velocity component, $u_{i+1/2,j}$, at the edge $(i + 1/2, j)$ of cell d is computed in the same manner; here, however, there is no need to extrapolate this velocity component, as it should have already been computed in the pressure projection step for the fluid cell e ($\phi < 0$), incorporating the free-surface boundary condition (see Equation 3.26). As another example, cell g ($\phi > 0$) has no fluid cells ($\phi < 0$) adjacent in the x-direction (both neighbouring cells i and h have $\phi > 0$). So, the velocity components of cell g are directly copied from the fluid cell j in the z-direction; the contribution from the x-direction is omitted. It should be also noted that in practice, a small number, which in this thesis is set to $1.0 \times 10^{-6}\Delta x$, is added to the denominator of Equation 3.41 in order to ensure numerical stability.

After the above procedure is completed, a correction step is then conducted. Here, the above procedure is repeated in cells from the first band near the free surface to the second band (and then to other further bands if necessary) as shown in Figure 3-6. But, at this time, cells that have $\phi \geq 0$ and already extrapolated velocities are considered by their adjacent cells. For example, previously, when extrapolating velocities for cell g (see Figure 3-6), the contribution from the x-direction is neglected

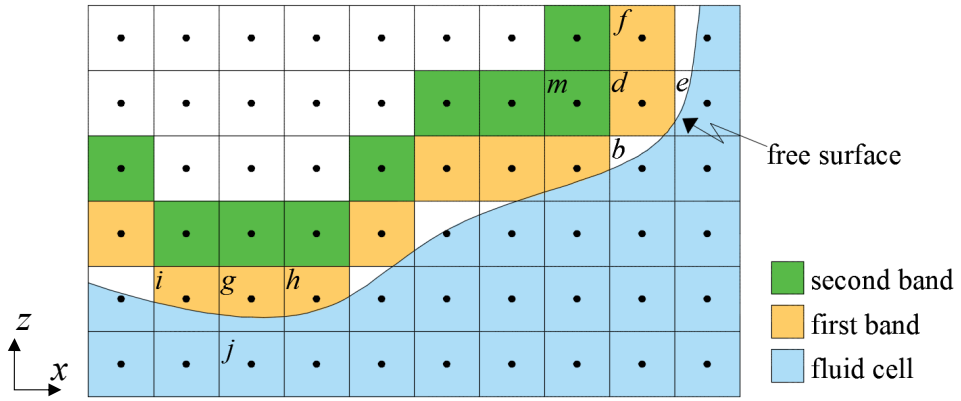


Figure 3-6: Schematic showing the configuration of a fluid body and the bands of cells where velocities are extrapolated. The black dots are at the centres of each cell.

when solving Equation 3.39, due to that cells i and h both have $\phi > 0$. However, in this correction step, cells i and h should already have extrapolated velocities and are now considered for the repeated velocity extrapolation in cell g using Equation 3.39. Note that in the correction step, when selecting a cell from the two neighbouring cells in one direction, the following two rules are applied: (1) the cell of the lower band is chosen (note that fluid cells with $\phi < 0$ are considered as being in the lowest band in this sense); (2) if both cells belong to the same band, then the one with a smaller SDF value (i.e. closer to the free surface) is chosen. For example, cells i and h both belong to the first band as shown in Figure 3-6; therefore, the one with a smaller SDF value is chosen as the x-direction component when solving Equation 3.39 for cell g.

3.2.4 Particle redistribution scheme

One problem inherent in the PIC method, as reported in Brackbill (1988); Edwards and Bridson (2012) and many others, is that the particle distribution tends to become disordered over time. This may lead not only to low accuracy in terms of velocity transfer between particles and grid, but also to particle clustering and hence unphysical voids in the fluid region that could be troublesome for the computational cells around. For a review of this particle disordering issue within PIC, see Section 1.1. In order to achieve a better (i.e. more regular) particle distribution, a technique is used in the current model to slightly shift the particles; this is described as follows.

A particle redistribution scheme is developed based on that proposed in Ando et al. (2012). After a certain number of time-steps, the particles are slightly shifted; the idea being that when two particles get too close, they are shifted away from each other, along their connection line. The new position of a particle is expressed as:

$$\boldsymbol{x}^{new} = \boldsymbol{x}^{old} + \Delta\boldsymbol{x}, \tag{3.42}$$

where $\Delta\boldsymbol{x}$ is the particle displacement (note this is different from the cell length Δx). Considering the isotropic neighbouring particles, the displacement is averaged using the cubic spline kernel function (see Equation 3.34):

$$\Delta\boldsymbol{x}_i = -\gamma_s \sum_j \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} W_i(\boldsymbol{x}_i - \boldsymbol{x}_j, k_h)\,, \tag{3.43}$$

where γ_s is a user-defined constant and could be problem-specific. It is found that the simulation results are not unduly sensitive to the value of this coefficient; however, the coefficient should not be too large or too small as it may cause a large increase in CPU time or even cause the scheme to fail to converge. In particular, fluid particles may cross solid wall boundaries if too large a value is used for γ_s. The recommended values are $0.015\Delta x$ in two spatial dimensions and $0.05\Delta x$ in three spatial dimensions, respectively. Note that Equation 3.43 sums up the neighouring particles only when the distance between the two particles is less than a user-defined value, which in the current model is set to a half of the cell length.

The redistribution scheme is repeated until a termination criterion is reached. For 2D problems, the termination is based on the average covariance of particle position, which is computed, for particle i, as:

$$C_i = \frac{\sum_j (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_i)(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_i)^T W_i(\bar{\boldsymbol{x}}_i - \boldsymbol{x}_j, k_h)}{\sum_j W_i(\bar{\boldsymbol{x}}_i - \boldsymbol{x}_j, k_h)}\,, \tag{3.44}$$

where

$$\bar{\boldsymbol{x}}_i = \frac{\sum_j \boldsymbol{x}_j W_i(\boldsymbol{x}_i - \boldsymbol{x}_j, k_h)}{\sum_j W_i(\boldsymbol{x}_i - \boldsymbol{x}_j, k_h)}\,. \tag{3.45}$$

When the sum of the average covariance of all particles converges to within a tolerance of 10^{-3}, the redistribution scheme is stopped. Note that Equation 3.44 requires a reuse of the indexes of the neighbouring particles that are involved in solving Equation 3.45. This increases the burden of the requirement for memory storage,

especially for simulations in three spatial dimensions. Thus, for 3D problems, the current model uses another termination criterion:

$$C = \frac{\sum_i \|\Delta \boldsymbol{x}\|}{\sum_i 1.0} < 0.01\gamma_s \ . \tag{3.46}$$

The idea being that the redistribution scheme is stopped when the average displacement of the particles becomes small enough as the particles tend to become more regularly distributed during the course of the redistribution. It is noted that in 3D computations a maximum 500 iterations restriction is applied to save on CPU cost in case that the converging rate is low. Also, in the current 3D simulations, the particle redistribution scheme is only applied at those fluid particles around fixed structures, mainly to reduce the unphysical voids around structures that may affect the calculation of the fluid force on the structures.

During the particle redistribution, the particles around the free surface are fixed, so that the shape of the free surface is not affected. Similar ideas have also been used in the SPH method (see e.g. Lind et al. (2012) and Colagrossi et al. (2012)). Furthermore, in the ghost cells, fixed ghost particles are also seeded (4 particles per cell in two spatial dimensions and 8 particles per cell in three spatial dimensions) to prevent particles from crossing the domain boundary. Similarly, if there are any other structures within the fluid region, the particles (or boundary points) that represent the structure surfaces must also be fixed.

In order to avoid affecting the characteristics of the velocity field that is carried by the particles during the redistribution, the velocity field is first mapped from the particles to the grid before the redistribution (using Equation 3.31), and after that it is interpolated back onto the particles (using Equation 3.11). Alternatively, the redistribution can be applied after the divergence-free velocity field is found on the grid (i.e. after solving Equation 3.8) and then use only the velocity itself on the grid to update the particle velocity (i.e. $c = 0$ in Equation 3.9). In either way, the cost, however, is that additional numerical diffusion is introduced, especially when the redistribution scheme is applied very frequently. Therefore, in the current solver the redistribution scheme is applied every N_r time-steps, and unless otherwise stated N_r is set to 30; the effect of this coefficient is demonstrated in Section 3.3.

Figure 3-7 compares snapshots of simulations with and without the particle redis-

tribution scheme for wave overtopping over a low crest structure that is described in chapter 5. It can be seen that the particle redistribution scheme effectively reduces the disorder of particle distribution, particularly around the structures. It is also observed that the free-surface location is not disturbed when the redistribution scheme is used.

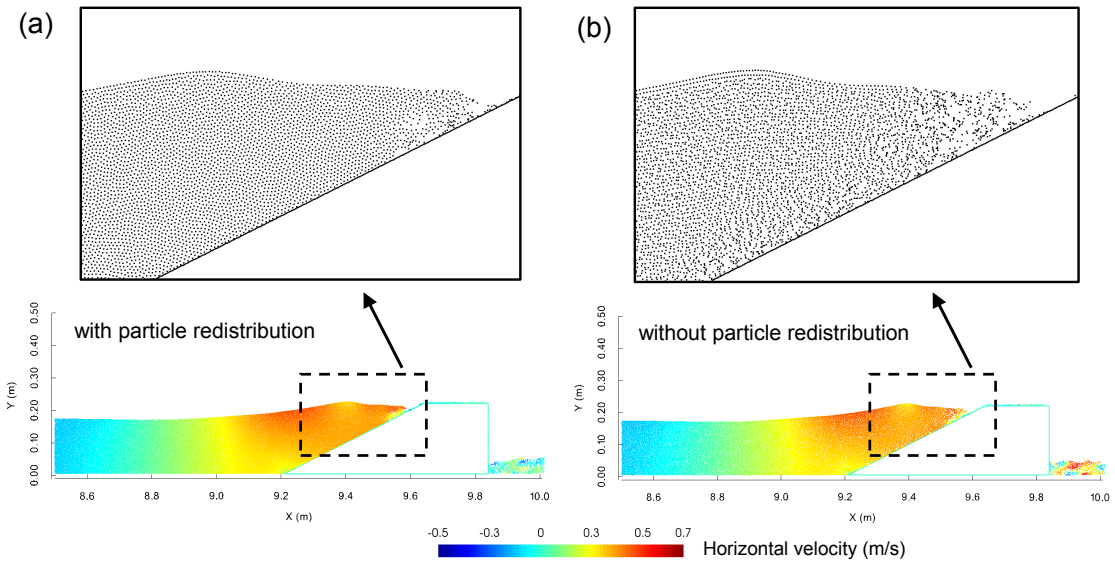


Figure 3-7: Comparison of snapshots simulated with and without the particle redistribution scheme for wave overtopping over a low crest structure.

3.2.5 Numerical accuracy and stability

Numerical accuracy analysis

The numerical solutions of CFD models are approximations of the exact flow. It is well known that one source of the numerical errors is the discretisation and solution of governing equations both in time and space. For the current model, the accuracy in time is first-order. This is primarily due to the explicit Euler time-advancing scheme. For the accuracy in space, different components of the solver are studied: (i) the transfer of velocity from the particles to the grid is first-order accurate; (ii) the discretisation of PPE is second-order accurate; (iii) the interpolation of velocity from the grid to the particles is fourth-order accurate; (iv) the advection of particles is third-order accurate. Thus, based on this error source, the overall accuracy of

the current model in space is determined by the crucial mapping from the scattered particles to the uniform grid, which is of first-order accuracy formally.

Time step and numerical stability

In order to ensure numerical stability, some necessary conditions have to be satisfied to determine the time step Δt. Following Markham and Proctor (1983), in the current solver it is required that any particle should not move across more than one cell in each direction in one time step. Thus, in the uniform grid the variable time step is restricted by:

$$\Delta t \leqslant \frac{\Delta x}{|\max(\boldsymbol{v}_p)|}. \tag{3.47}$$

In the above equation, determining $\max(\boldsymbol{v}_p)$ is not trivial as it first seems. This is because $|\max(\boldsymbol{v}_p^{n+1})|$ could be greater than $|\max(\boldsymbol{v}_p)|$ due to the effect of body forces. Thus, a conservative form of $\max(\boldsymbol{v}_p)$ is to consider the body forces:

$$|\max(\boldsymbol{v}_p)| = |\max(\boldsymbol{v}_p^n)| + \Delta t|\boldsymbol{f}|. \tag{3.48}$$

Considering the upper bound of Equation 3.47 and substituting it into Equation 3.48, the contribution of body forces to $|\max(\boldsymbol{v}_p)|$ can be further modified to $(\Delta x|\boldsymbol{f}|)^{\frac{1}{2}}$. Hence, Equation 3.48 can be rewritten to:

$$|\max(\boldsymbol{v}_p)| = |\max(\boldsymbol{v}_p^n)| + (\Delta x|\boldsymbol{f}|)^{\frac{1}{2}}. \tag{3.49}$$

Substituting Equation 3.49 into Equation 3.47, we finally have:

$$\Delta t \;=\; \frac{\Delta x}{|\max(\boldsymbol{v}_p^n)| + (\Delta x|\boldsymbol{f}|)^{\frac{1}{2}}} C_F, \tag{3.50}$$

where $C_F \leq 1$ is the Courant number.

Another restriction on the numerical time step comes from the viscosity term of the momentum equation. This is expressed as (Griebel et al., 1998):

$$\frac{C_\nu \nu \Delta t}{\Delta x^2} \leq 1. \;, \tag{3.51}$$

where C_ν equals 2, 4 and 6 in one, two and three spatial dimensions, respectively. This constraint could become dominant when the flow viscosity is high and/or the

mesh is fine. In practice, simulations of water-wave type problems in this study show that the time step is controlled by the Courant number, i.e. Equation 3.50.

3.2.6 Numerical algorithm

The solution procedures of the "full particle" PIC model presented in this chapter is summarised in Figure 3-8. The variables adjusted at each step of a computational cycle are also indicated.

3.3 Test cases

In this section, the "full particle" PIC model developed in this chapter is examined by two benchmark tests of free-surface flows: 1) standing wave; 2) dam break flow. Comparisons between results of the current model and analytical solutions or physical experimental data are made to verify the numerical model.

3.3.1 Standing wave

A standing wave in a 2D wave tank is simulated in this section. A cosine shaped free-surface profile is given as the initial setup; this is shown in Figure 3-9. The tank length, D, and the wave length, L, were both set to 20 m, and the water depth, h, was 10 m, which gives a deep water condition, as $h/L = 0.5$. The initial wave height, H, was 0.6 m, resulting in $H/L = 0.03$. So, linear wave theory was used to provide analytical solutions for comparison. The Courant number, controlling the variable time step, was 0.5 in this case.

A grid refinement study, on the measurement of wave profile at the middle point of the tank, i.e. $x = 0.5D$, was carried out with four different grid sizes: A) $\Delta x = \frac{1}{3}H$ = 0.2 m; B) $\Delta x = \frac{1}{6}H$ = 0.1 m; C) $\Delta x = \frac{1}{12}H$ = 0.05 m; D) $\Delta x = \frac{1}{24}H$ = 0.025 m. The results are plotted in Figure 3-10. It can be seen that the results are almost identical before $t \approx 13$ s, after which discrepancy occurs as shown in the

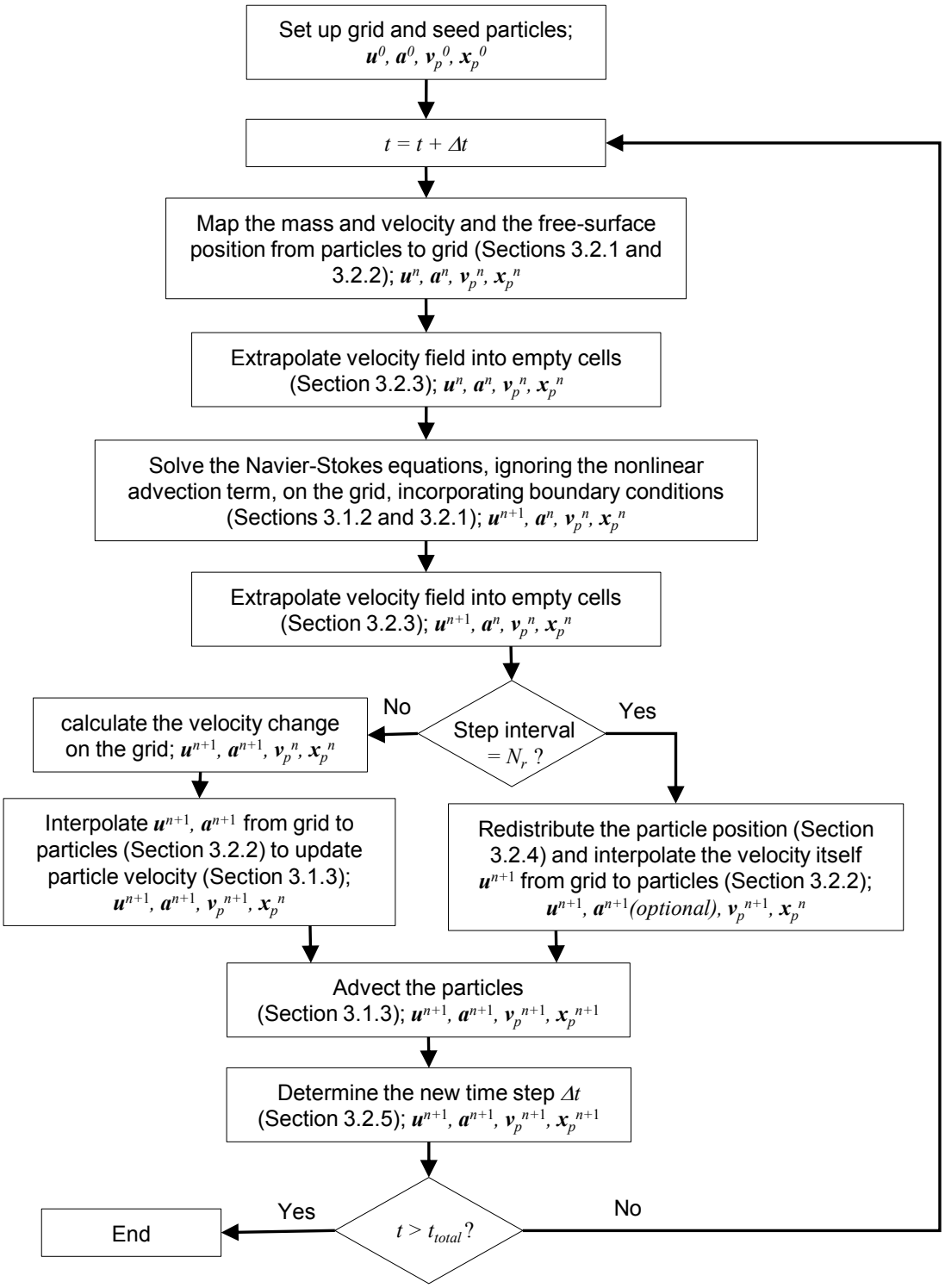


Figure 3-8: Flow chart of the solution procedure of the present "full particle" PIC model.

enlarged area. It is seen that the predicted wave period, T, run on a coarse grid tends to become larger than that run on a fine grid as the simulation progresses. This is likely because the relatively large numerical diffusion induced by the coarse grid tends to gradually dissipate the kinetic energy of the wave and decrease the natural frequency of this vibration system. Focusing on the predicted wave profiles between $t = 13$ s and 19.8 s and taking that of the grid size $\Delta x = 0.025$ m (the finest grid above) as the reference solution, the root mean square errors (RMSE) of the wave profiles of the grid sizes $\Delta x = 0.05$ m, 0.1 m and 0.2 m, normalised by the wave height, are approximately 1.1%, 3.7% and 9.3%, respectively. Therefore, it is seen from both the RMSE and Figure 3-10 that in the presented time period of simulation, the results become insensitive to the grid size when the grid is fine enough. In particular, when using the grid size $\Delta x = 0.05$ m, the result is nearly identical to that of using $\Delta x = 0.025$ m. This indicates that the numerical model is convergent when the grid size $\Delta x = 0.05$ m is used. This grid size, resulting in 402×222 cells and approximately 320,000 particles, was thus employed to produce results for comparison with analytical solutions.

Figure 3-11 shows a comparison of the wave profile at $x = 0.5D$ between the numerical and the analytical solutions. It is seen that good agreement is achieved; the RMSE normalised by the wave height is approximately 3.1%. It is also noteworthy that both the wave amplitude and phase are well predicted by the current model. Figure 3-12 presents the dynamic wave pressure along the water depth at $x = 0.5D$ for two different time instants. The left panel shows the pressure after one wave period, and the right panel plots the pressure after one and a half wave periods. It is seen from Figure 3-12 that the numerical results at both time instants agree well with the analytical solutions. The RMSE of the two results normalised by $\rho g H/2$ are 3%

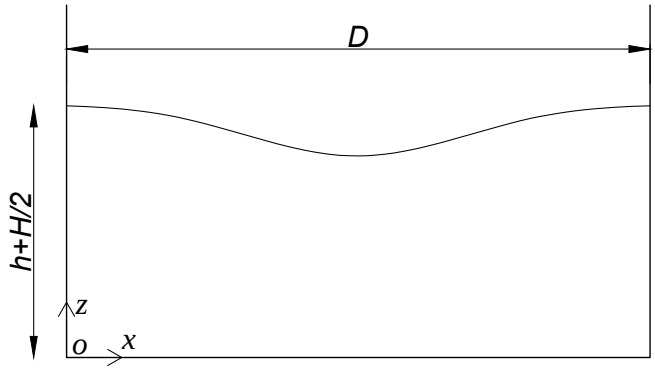


Figure 3-9: Numerical setup for standing wave test.

and 2.3%, respectively, which hints that the PPE is well solved in the current model for the pressure. Since the pressure is well computed, the mass conservation can be expected to be preserved to a reasonable extent. This is confirmed by examining the total mass of the fluid body as shown in Figure 3-13, from which it can be seen that only roughly 0.01% of the mass is slowly lost during the course of 20 seconds of simulation, and this figure can be expected to be smaller when a finer grid is used.

3.3.2 Dam break

This test case concerns the capability of the current model of dealing with violent free-surface flows. This benchmark test is widely used to test many numerical models such as those based on the SPH method (see e.g. Hughes and Graham (2010) and Marrone et al. (2011)). A column of water of width W and height H is contained in a tank of width D and instantaneously released, simulating the collapse of a water

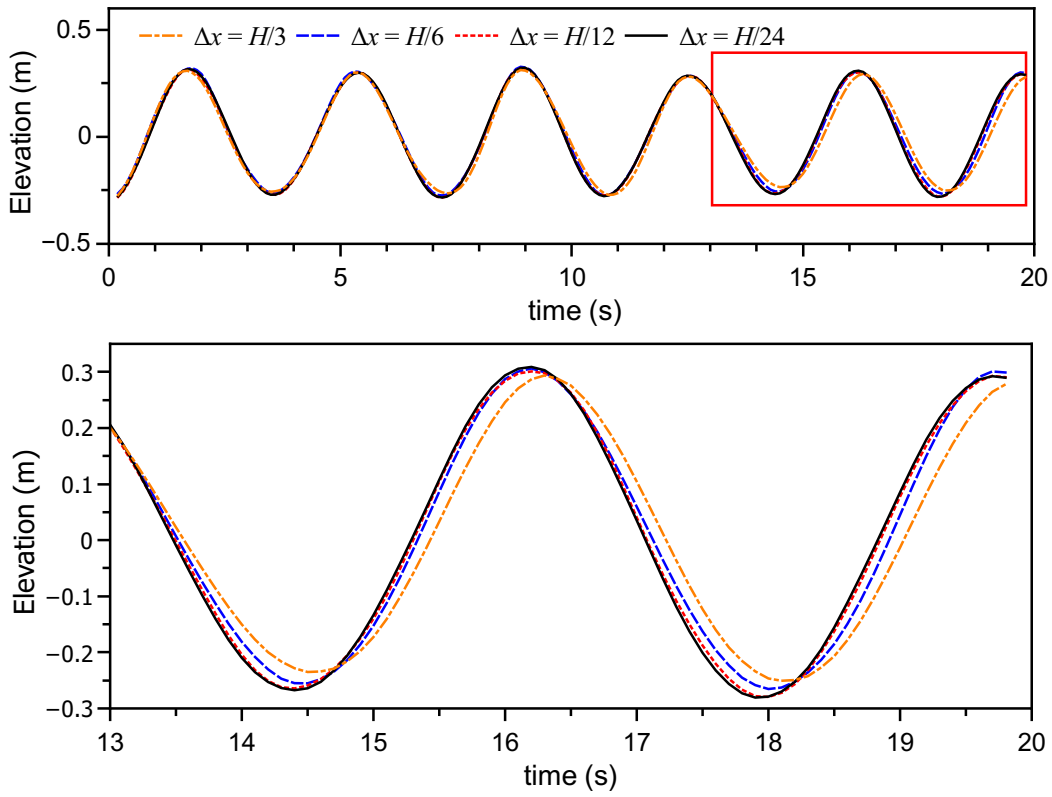


Figure 3-10: Grid refinement study on the prediction of wave profile at $x = 0.5D$. Upper panel: full time length of the simulation (0–20 s); lower panel: enlarged view of the area marked by the box shown in the upper panel.

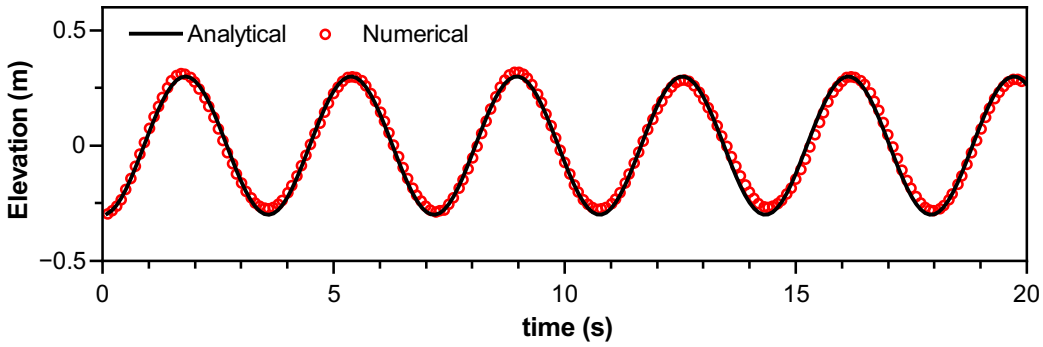


Figure 3-11: Comparison of the time histories of wave profile at $x = 0.5D$.

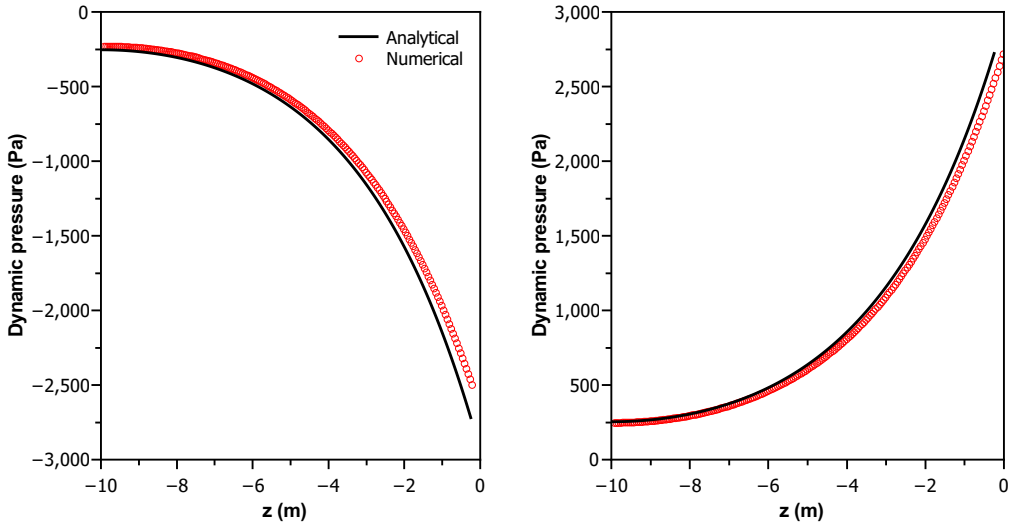


Figure 3-12: Dynamic wave pressure as a function of water depth at $x = 0.5D$ at time instants 3.61 s (left panel) and 5.39 s (right panel).

dam. The water flows across the horizontal dry bed and eventually impacts a vertical wall. The initial setup is plotted in Figure 3-14.

In this study, two different setups related to the sizes of the water column and the tank were used. For the first one, this was $W = 2H$ and $D = 5.366H$. The grid size used for this case was $H/40$, resulting in 217×150 grid cells with about 12800 particles. The Courant number was set to 0.5.

Figure 3-15 shows snapshots of the flow profile, represented by the particles, at different time instants. The pressure field is also interpolated from the grid to the particles for visualisation. In addition, two blending coefficients (see Equation 3.9),

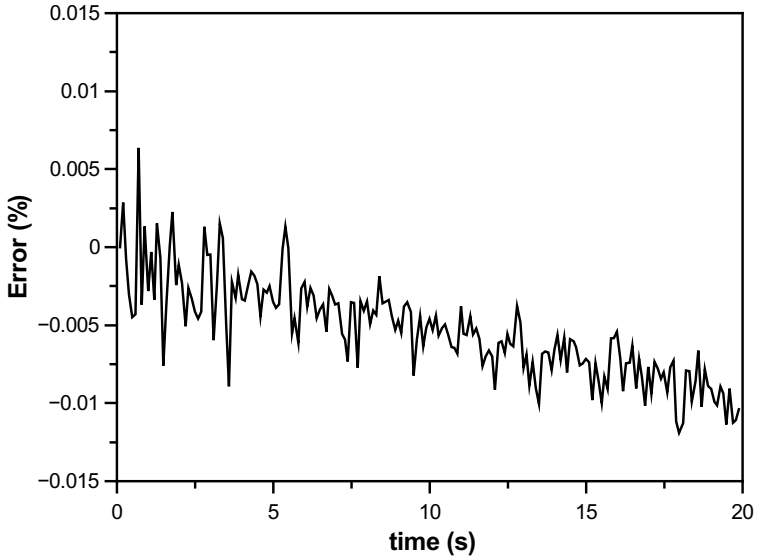


Figure 3-13: Numerical errors in maintaining the mass of fluid body in the standing wave case.

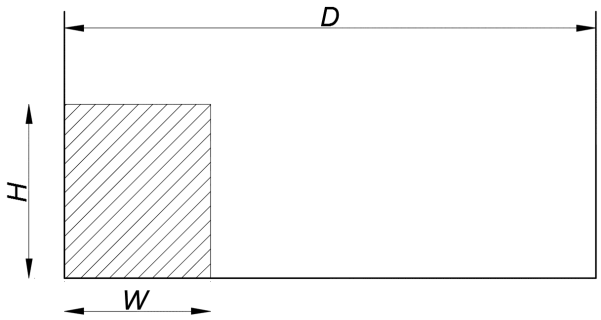


Figure 3-14: Numerical setup for dam break test.

c, were tested, for which the results at the same time instants were compared. It is clearly seen that the flow, modelled with $c = 0.0$ (i.e. the "classical" PIC), is more viscous as expected, due to the numerical dissipation from the direct velocity transfer. In contrast, the flow profile, as well as the pressure field, run with $c = 0.96$, are very similar to those obtained by the state-of-the-art δ-SPH model (c.f. Figure 3 of Marrone et al. (2011)). Also, it is worth mentioning that due to the fact the pressure is solved at the grid in the current model, the present pressure field appears to be smoother than that from the meshless δ-SPH model.

For the second case, the setup was $W = H/2$ and $D = 2H$, following Greaves (2006) and Hughes and Graham (2010). Other numerical settings were the same as

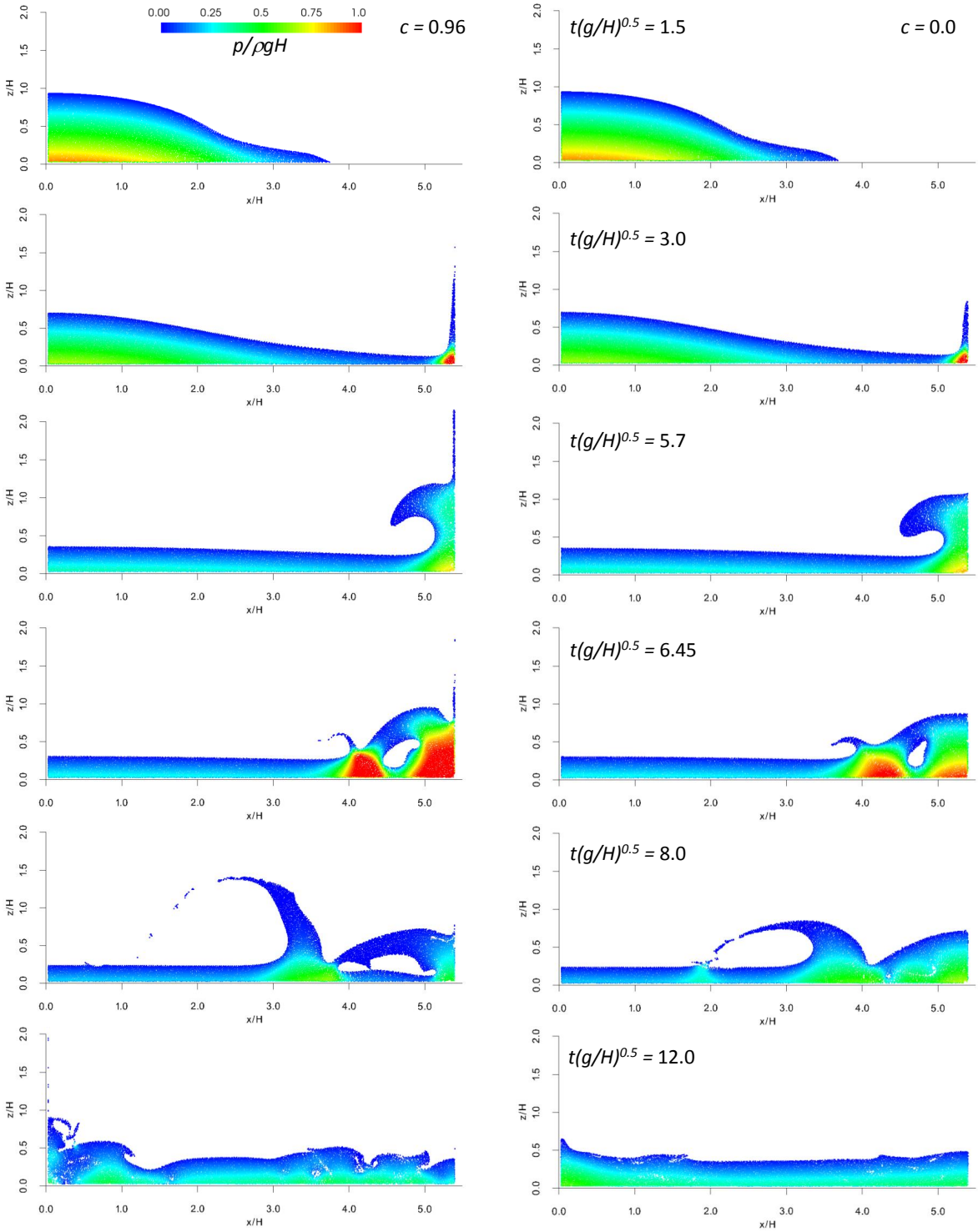


Figure 3-15: Snapshots of the dam break test to compare the influence of the blending coefficient $c = 0.96$ (left column) and $c = 0.0$ (right column), at the same time instants.

those used in the first case above. For this case, numerical results of the adaptive hierarchical grid-based method (Greaves, 2006) and the meshless WCSPH method (Hughes and Graham, 2010), as well as physical experimental data, were used for comparison.

Figure 3-16(upper panel) compares the leading front position of the dam break flow, between the numerical predictions and the experimental observation. It is seen that the results from the current model are in good agreement with those from other numerical models, despite that all the numerical results present phase differences compared with the experimental observation. This discrepancy, as noted in Greaves (2006), is probably due to that the physical experiment encountered difficulties in capturing the front of the flow in such a fast flow scenario. Note that in the current simulation, tracking the leading front of the flow motion was carried out at the particle level. Figure 3-16(lower panel) presents a study on the sensitivity of the blending coefficient, c, and the time-step interval, N_r, of the particle redistribution scheme. It is seen that the time histories of the front position are almost identical for the different values of c used, which implies that the influence of this coefficient is very small when it is close to, but less than, unity. Thus, in this thesis, c was set to 0.96 for all test cases unless otherwise stated. It is also seen that the value of N_r has a very slight effect on the results when it is around 10 to 30, although it appears that applying the redistribution scheme more frequently (i.e. using a smaller value of N_r; e.g. $N_r = 10$) introduces more numerical dissipation that slows down the flow motion slightly.

Figure 3-17(upper panel) compares the time history of the height of the water column at the left wall boundary. The column height is also recorded by the particles. It is observed that the present result is in close agreement with those from other numerical models, as well as the experimental data. This indicates that the advection of particles around both the free surface and the domain boundary is well computed in the current model. Figure 3-17(lower panel) shows a comparison of the column height for two grid-to-particle interpolation schemes (see Section 3.2.2); one of them is the fourth-order accurate WENO scheme, the other is the second-order accurate conservative scheme. Note that, as mentioned in Section 3.2.2, for the former, grid values (i.e. fluid velocity) beyond the ghost cells are required and they are simply set to zero in the current model, while the latter does not required so. It can be observed that the results of the two schemes are almost indistinguishable, hinting

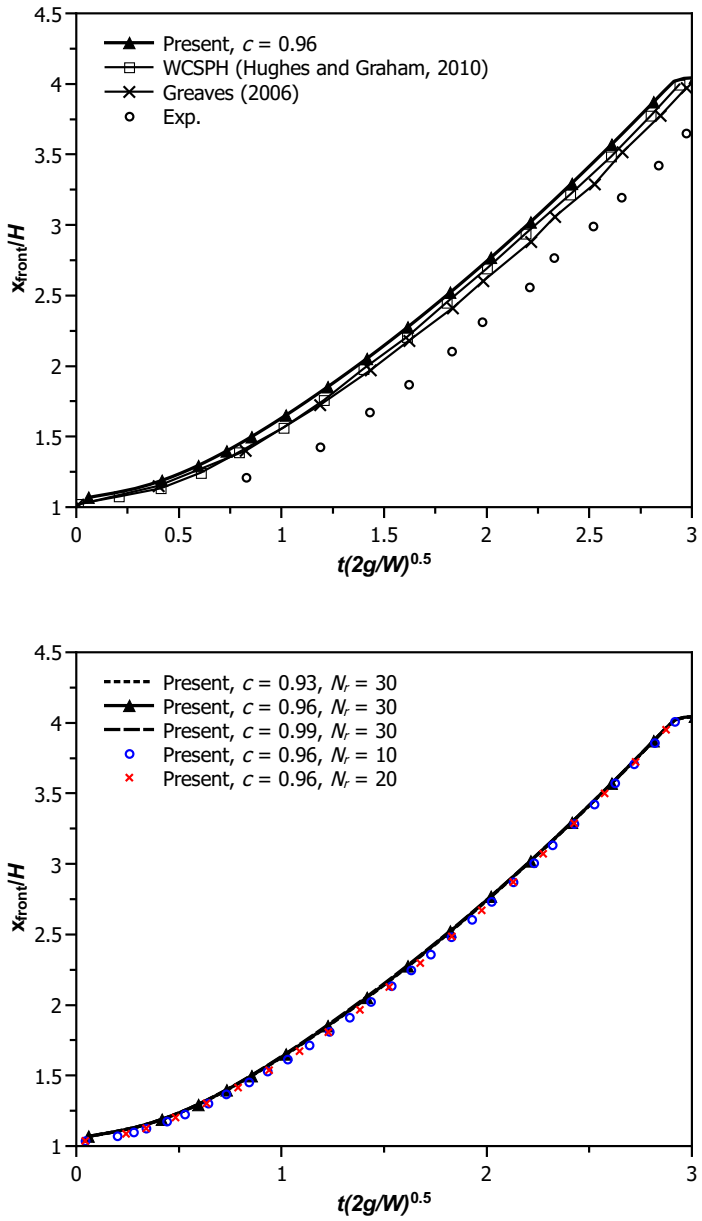


Figure 3-16: Comparison of the dam break front position. Upper panel: comparison for the present numerical results ($c = 0.96$), results from other numerical models and the experiment; lower panel: comparison for the present results run by different values of c and N_r.

that the above-mentioned settings used for the WENO scheme around the domain boundary are acceptable.

Finally, it is worth noting that for the current model all the test cases in this chapter were run on an Intel Core i7-4600U (CPU @2.10Ghz∼2.70GHz) laptop with

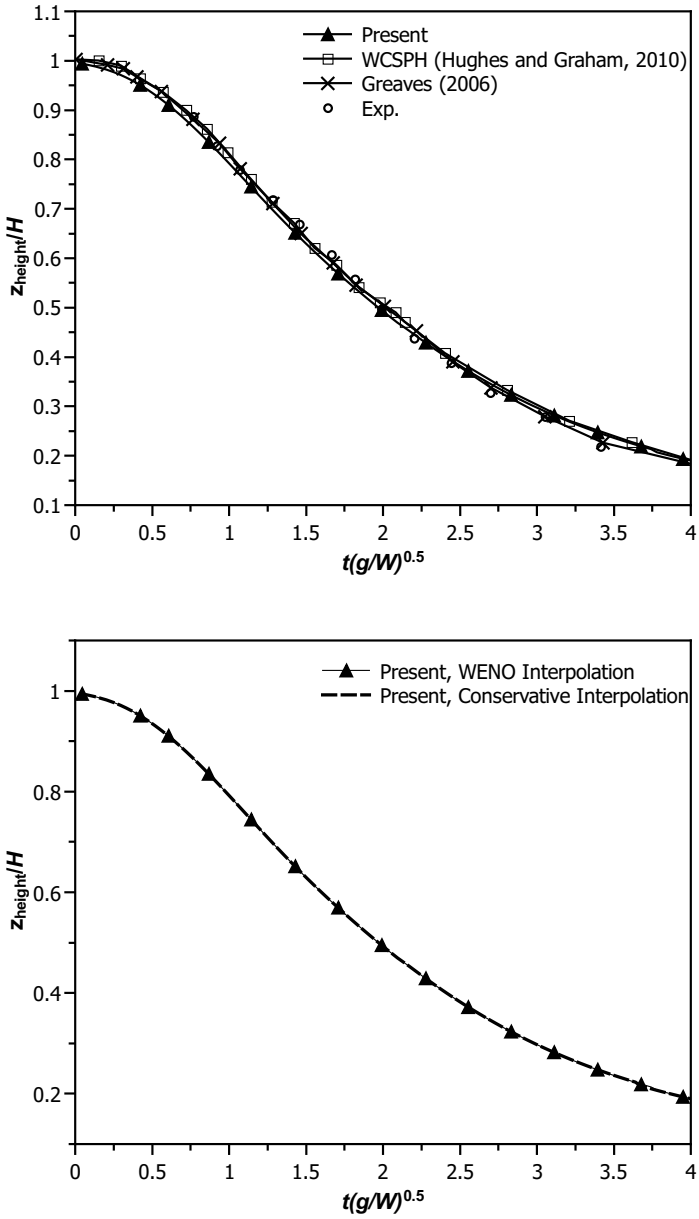


Figure 3-17: Comparison of the water column height at the left wall boundary. Upper panel: comparison for the present numerical results, results from other numerical models and the experiment; lower panel: comparison for the present results run by different schemes for interpolating the velocity from grid to particles for advecting the particle positions.

8Gb RAM employing a Windows OS. For the dam break test with the geometry of $W = H/2$ and $D = 2H$, the CPU time of the present "full particle" PIC model for $t(g/W)^{\frac{1}{2}} = 4.5$ dimensionless time of simulation was 2.4 min when employing approximately 12,800 particles. In contrast, the WCSPH model and Incompress-

ible SPH (ISPH) model both described in Hughes and Graham (2010) require 73 min and 88 min, respectively, for such a simulation with 5000 fluid particles. The present model is thus, in this case, significantly more efficient than the SPH models of Hughes and Graham (2010). This is primarily because in SPH models the number of interactions with neighbouring particles for each particle can be up to 100 in 2D simulations and 200–500 in 3D simulations (Violeau and Rogers, 2016), which is far more than the number of neighbouring cells for the current finite difference or finite volume scheme.

3.4 Summary

The techniques used to develop a "full particle" PIC model for solving the incompressible NS equations for single-phase free-surface flows were detailed in this chapter. In the model, both Eulerian grid and Lagrangian particles are employed for the computation. The particles carry all the material information of the fluid body and are used to solve the nonlinear advection term and track the free surface, while the underlying grid is solely employed for computational convenience for solving the non-advection terms in the NS equations. The solution scheme is divided into two stages: the Eulerian stage and the Lagrangian stage. During the Eulerian stage, the NS equations are resolved, ignoring the nonlinear advection term, which is handled by moving the particles in a Lagrangian manner during the Lagrangian stage. Meanwhile, different interpolation techniques are utilised to transfer the information back and forth between the scattered particles and the uniform staggered grid.

The "full particle" PIC model is validated through a few benchmark tests of free-surface flows, including standing wave in a tank and more violent dam break flow. Results show that the current model is capable of dealing with small amplitude standing waves, as well as more violent dam break flows, with reasonable accuracy, flexibility and efficiency. Also, it is found that the current model is convergent, with the mass conservation well preserved and the numerical dissipation kept within an acceptable level. In the test cases, the influences of some user-specific coefficients are tested and appropriate values for them are recommended. For example, the blending coefficient, c, used to weight the velocity interpolation for particles between

"classical" PIC and "full particle" PIC, is found to be insensitive when it is close to but less then unity.

The next step is to develop algorithms that are suitable for the current PIC framework, for investigating fluid-structure interactions, especially those occurring in the coastal and offshore engineering field. These are described in the following chapters.

CHAPTER 4

A DISTRIBUTED LAGRANGE MULTIPLIER METHOD BASED SCHEME FOR FLUID-SOLID INTERACTION: SIMULATING 2D FALL-PIPE ROCK DUMPING

In this chapter, an extension of the DLM technique proposed by Patankar (2001) is employed for dealing with fluid-solid interaction within the "full particle" PIC framework developed in Chapter 3. This tailored version of the DLM method allows the model to simulate freely movable rigid objects of arbitrary shapes within the fluid. As noted in Carlson et al. (2004), where the DLM method is also applied, the idea being that solids are initially modelled as if they were fluids, and then a correction on the velocity field, within the solid phase, is made to account for density differences and solid collisions. Finally, rigidity constraint is enforced within the solid phase to obtain final velocities for the solids. Detailed implementations of this technique are described in the following sections, followed by test cases involving the modelling of fall-pipe rock dumping in two spatial dimensions.

The targeted placement of rock onto the sea bed is a common solution for the protection of offshore pipelines and cables, and scour protection for windfarm structures. In deeper waters (50 m or more) to ensure the accuracy of this placement, rock is dumped via a fall-pipe–a long, vertical, semi-open, flexible fall-pipe of typically around 1 m in diameter. Usually the pipe consists of a string of bottomless, heavy buckets which extends from the work vessel to a position just above the seabed where the material is to be placed (IADC, 2012). Rock (typically several inches in diameter) is placed into the top of the pipe from a conveyor belt and exits the pipe over the required location. Rock fall-pipes may display clumping behaviour depending on the volume concentration, which is common to many granular systems in pipes (see e.g. Raafat et al. (1996); Nakahara and Isoda (1997)). This clumping behaviour may cause the failure of pipes and limit productivity. Therefore, the main focus

of this chapter is on the development and application of the numerical model as a starting point for attempting to capture and understand the physical processes of rock dumping in fall-pipes. So, the numerical model could work as a low cost basis for optimisation and improvement of the productivity of fall-pipe rock dumping. As an initial study, the pipes here are simplified to be rigid and fixed in water, and since the simulation is in two spatial dimensions, the rocks are modelled using 2D plates.

4.1 Governing equations

In the presence of solids, the computational domain Ω is divided into fluid phase and solid phase, which are denoted by Ω_f and Ω_s, respectively. Figure 4-1 shows an overview of the fluid domain with an immersed solid. The boundary of a solid is represented by a set of points with an equal distance between two successive points, and these points are referred to as the boundary points hereafter. A rectangular region covering the solid phase is also indicated by dashed lines in Figure 4-1. This rectangular region is related to the implementation of the tailored DLM method discussed in the following section. Note that in the solid phase, particles are also seeded, with the same configuration as those in the fluid phase. These particles inside the solid phase are used for solving the solids as if they were fluids as mentioned above. Here, the detection of whether a particle or a grid point lies inside or outside a 2D solid object is straightforward; based on the INPOLY algorithm from the TELEMAC suite (see `http://www.opentelemac.org/`), the approach first introduces a ray from the particle or the grid point, and then finds its intersections with the line segments between any two successive boundary points; if the ray cuts the solid with an odd number intersections then the point is regarded as being inside the solid object.

Recalling the incompressible NS equations for the fluid, the system of equations governing the fluid and the rigid solid are expressed as:

$$\nabla \cdot \boldsymbol{u} \;=\; 0 \quad \text{in } \Omega_f, \tag{4.1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \boldsymbol{f} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} \quad \text{in } \Omega_f, \tag{4.2}$$

and:

$$\mathbf{D}\left[\boldsymbol{u}\right] \;=\; 0 \quad \text{in } \Omega_s, \tag{4.3}$$

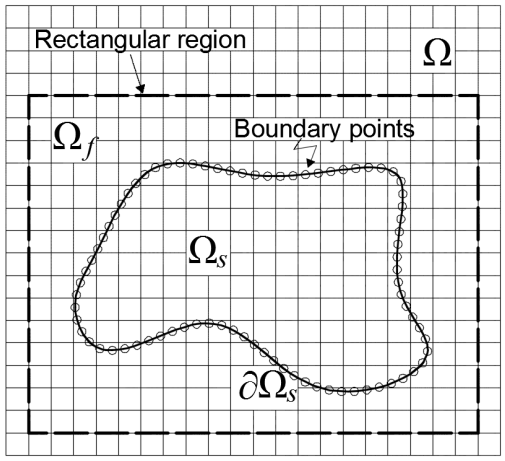


Figure 4-1: Schematic showing the computational domain Ω and embedded solid Ω_s and fluid Ω_f phases.

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \boldsymbol{f} - \frac{1}{\rho_S}\nabla p + \nabla \cdot \boldsymbol{\Pi} \quad \text{in } \Omega_s, \tag{4.4}$$

with the added boundary conditions on moving solid boundaries:

$$\boldsymbol{u} = \boldsymbol{U}_b \text{ and } (\boldsymbol{\Pi} - p\boldsymbol{I}) \cdot \boldsymbol{n} = \boldsymbol{T} \quad \text{on } \partial\Omega_s(\boldsymbol{x}, t), \tag{4.5}$$

where Equation 4.3 and Equation 4.4, having similar forms to Equation 4.1 and Equation 4.2 for the fluids, are the rigidity constraint and momentum equation for the solids. As noted in Patankar et al. (2000), similar to Equation 4.1, which enforces the fluid incompressibility and gives rise to the pressure p as a Lagrange Multiplier, Equation 4.3, where $\mathbf{D}[\boldsymbol{u}] = \left[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right]/2$, enforces the deformation-free velocity field within the solid phase and leads to another Lagrange Multiplier $\boldsymbol{\Pi}$ as an extra stress tensor in addition to the pressure field p in Equation 4.4. The rigidity constraint also eliminates any diffusion term within the solid phase. Also, in the above boundary conditions, $\boldsymbol{U}_b$ is the velocity field on the solid boundary; ρ_S is the solid density; the traction force of the fluid on the solid, denoted by $\boldsymbol{T}$, is the sum of the projected viscous stresses and pressure; $\boldsymbol{I}$ is the identity tensor and $\boldsymbol{n}$ is the outward normal vector on the solid surface. Note however that Equation 4.3, $\boldsymbol{\Pi}$, and the boundary conditions, in the current model are not solved for directly but implicitly enforced through the approach proposed in Patankar (2001); this is detailed in the following sections.

4.2 Numerical scheme

This section details the implementation of the tailored version of the DLM techniques proposed in Patankar (2001) for fluid-solid interaction, within the current "full particle" PIC framework. In particular, the solution procedure involves three steps: i) solids are first solved as if they were fluids; ii) within the solid phase, a correction on the velocity field is made to account for density differences (between the fluids and solids) and any momentum transfer due to solid collisions; iii) the rigidity constraint is enforced to obtain the final velocities of the solids.

For the first step, Equation 4.1 and Equation 4.2, excluding the advection term, are first imposed on the entire background grid domain, using the methods for the Eulerian stage as described in chapter 3. Note that the viscosity term in Equation 4.2 is not applied within the solid phase, as it is eliminated by the rigidity constraint. After the divergence-free velocity field, here denoted by $\boldsymbol{u}^{n+1/2}$, is obtained via the pressure projection step (see Equation 3.8), the velocity field within a rectangular region (see an example in Figure 4-1), covering the solid phase, is tentatively advected (on the grid) to complete the first step. The resulting velocity field within this region is represented by $\dot{\boldsymbol{u}}$, and this is achieved by solving the advection term in the conservative form, which is expressed, using the chain rule and the divergence-free condition, as:

$$u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} = \frac{\partial (u^2)}{\partial x} + \frac{\partial (uw)}{\partial z}\,; \tag{4.6}$$

$$u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z} = \frac{\partial (uw)}{\partial x} + \frac{\partial (w^2)}{\partial z}\,. \tag{4.7}$$

The above advection terms are solved by the method proposed in Seibold (2008), which uses a first-order accurate Euler integration in time and a second-order accurate differencing scheme in space that combines centered and upwind differencing.

The next step is to account for the effect of density differences between fluids and solids, and any momentum transfer due to solid-solid interaction. Within the rectangular region mentioned above, this velocity correction step is conducted according to Patankar (2001):

$$\hat{\boldsymbol{u}} = \dot{\boldsymbol{u}} + \theta \Delta t \boldsymbol{S}\ , \tag{4.8}$$

where,

$$\boldsymbol{S} = \boldsymbol{A}_c + \boldsymbol{r}_i \times \boldsymbol{\alpha}_c - (1 - \frac{\rho}{\rho_S})(\frac{D\dot{\boldsymbol{u}}}{Dt} - \boldsymbol{g}) \ , \tag{4.9}$$

where $\theta \in [0, 1]$ is the volume fraction of a computational cell that is occupied by solids, and the technique used to compute θ is described in Section 4.3; $\boldsymbol{\alpha}_c$ and $\boldsymbol{A}_c$ are the angular and the translational accelerations due to collisions between solids, and they are calculated following Patankar et al. (2000) (an example for calculating $\boldsymbol{A}_c$ is given in Section 4.5.2, Equation 4.23); $\boldsymbol{r}_i$ is the position vector with respect to the rotation centre of the solid. The third term at the RHS of Equation 4.9 accounts for the effect of density differences between fluids and solids. Note that the advection term in the material derivative here is also solved using the approach of Seibold (2008) as mentioned above.

Once Equation 4.8 is solved within the rectangular region covering the solid phase, the total momentum in the solid phase is correct and the rigidity constraint needs to be imposed in the third step. Patankar (2001) noted that because the momentum of the rigid body must be conserved, the unique velocity of the solid is simply the overall momentum divided by the solid mass:

$$\boldsymbol{U} = \frac{\int_{\Omega_s} \rho_S \hat{\boldsymbol{u}} \ dV}{M} \, , \tag{4.10}$$

and

$$\boldsymbol{\omega} = \frac{\int_{\Omega_s} \boldsymbol{r} \times \rho_S \hat{\boldsymbol{u}} \ dV}{I_s} \, , \tag{4.11}$$

where M and I_s are the mass and moment of inertial of the rigid solid; V is the volume of solid and in the discretisation form dV is actually the volume fraction of a cell occupied by a solid, i.e. θ in local cells; $\boldsymbol{U}$ and $\boldsymbol{\omega}$ are the final and unique translational and angular velocities of the solid, respectively; $\boldsymbol{r}$ is the position vector with respect to the rational centre. Thus, the velocity of any point on the grid within the rectangular region that covers the solid phase is computed as:

$$\hat{\boldsymbol{u}}_R = \boldsymbol{U} + \boldsymbol{r} \times \boldsymbol{\omega} \, , \tag{4.12}$$

and the final velocity field on the whole grid domain at the end of the Eulerian stage is obtained by:

$$\boldsymbol{u}^{n+1} = (1 - \theta)\boldsymbol{u}^{n+1/2} + \theta \hat{\boldsymbol{u}}_R \, . \tag{4.13}$$

Note that, despite the fact that the procedure described in this section is applied

to both the fluid and solid phases in the rectangular region, the computations for the fluid phase (i.e. $\theta = 0$) are not taken into account in the final calculation of the velocity in Equation 4.13. The fluid velocity is set equal to $\boldsymbol{u}^{n+1/2}$, corresponding to the divergence-free velocity field calculated in the first step above. Also, as a staggered grid is employed in the current solver, the velocity components are stored at different cell edges. Thus, following Ardekani et al. (2007), in Equation 4.10–Equation 4.13, the velocity components are separately computed based on their own velocity cells, i.e. cells surrounding velocities (see Figure 4-3). Because of this, the volume fraction θ is also computed separately; see Section 4.3 for the details. Furthermore, when multiple solids are simulated, Equation 4.8–Equation 4.13 are evaluated for each solid separately and simultaneously, within a rectangular region covering all the solids.

Once the correct velocities on the grid are found, the velocity change is computed both inside and outside the solid phase. The fluid particles are then advected using the method introduced in chapter 3 for the Lagrangian stage. The solids (i.e. the boundary points and the particles initially seeded inside the solid phase) are also advected at this stage using their final unique velocities, i.e. $\boldsymbol{U}$ and $\boldsymbol{\omega}$. In the current scheme, this uses a time integration that is only first-order accurate. It is possible to use a high-order method that further couples the fluid-solid motion (such as a predictor-corrector scheme). Such an approach, however, would be computationally very costly for the current solver.

Since the rigidity constraint is only applied within the solid phase, it is possible that the velocity field in the band of fluid cells immediately surrounding solids will not necessarily be divergence-free. One possible solution is to solve the fluid governing equations again, after the rigidity constraint is enforced, with the solids providing velocity boundaries for fluid computation. This approach however is obviously computationally very expensive. In practice it is found that it is not necessary to correct the velocities explicitly in these cells as they will be forced to be divergence-free at the next time-step, and thus the errors are bounded. It should also be noted that while the approach introduced in this section for handling fluid-solid interaction is momentum conserving and straightforward to implement, it could dissipate the kinetic energy of the solid, as Equation 4.10 and Equation 4.11 correspond to volume averaging operations.

Nevertheless, the DLM method proposed in Patankar (2001) and used here provides a solution to incorporating fluid-solid interaction in the "full particle" PIC flow solver that simply requires adding one simple fraction step to enforce the rigidity constraint in the solid domain after the fluid governing equations have been enforced on the entire computational domain. That is, after Equation 4.1 and Equation 4.2 are solved on the entire domain (note that the viscosity term is not applied in the solid domain), the momentum equation of the solids (Equation 4.4), except for $\mathbf{\Pi}$, is resolved. As mentioned above, $\mathbf{\Pi}$ is a Lagrange Multiplier resulted from the enforcement of the rigidity constraint (Equation 4.3), similar to the pressure p in the pressure projection method due to the divergence-free condition. Differently, however, here the rigidity constraint, so is $\mathbf{\Pi}$, are implicitly resolved by conserving the mass and momentum of the solids (Equation 4.10–Equation 4.12), rather than by solving a Poisson equation that is computationally expensive. Therefore, Equation 4.1, Equation 4.2 and Equation 4.10–Equation 4.12 are equivalent to Equation 4.1–Equation 4.4.

Due to the numerical errors mentioned above, there is a possibility that some fluid particles may enter solid objects after they have been advected. Here, a numerical technique is designed to project these fluid particles back into the fluid domain if this happens. Figure 4-2 shows a schematic of the concept of this numerical technique. Specifically, when a fluid particle is detected to be within a solid object (usually still around the boundary of the solid object as shown in Figure 4-2), the boundary points of the solid object around the fluid particle are checked to find the one that is closest to the fluid particle, which is B3 in the case presented in Figure 4-2. Then, the two successive boundary points B2 and B4 around B3 are checked to find the one that is closer to the fluid particle, which is B4 in this case. After that, the unit outward vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ along the connection lines between B3, B4 and the fluid particle are calculated respectively as shown in Figure 4-2, and the fluid particle is finally shifted by:

$$\Delta \boldsymbol{x}_i = \frac{\tau(\boldsymbol{a}+\boldsymbol{b})}{2i} \ , \ i = 1, 2, ..., \tag{4.14}$$

where τ is a user-defined coefficient (usually given around half the cell length). The fluid particle is repeatedly shifted by distances $\Delta \boldsymbol{x}_1$, $\Delta \boldsymbol{x}_2$, etc., so that the shifting distance is reduced at each shift. This procedure is terminated when the fluid particle is detected to be outside the solid object after each shift; otherwise, after certain number of shifts (four in the current model), the fluid particle is regarded as being "lost" and removed from the computational domain. It may be worth mentioning that in 3D simulations presented in chapter 6, this technique is modified. That is,

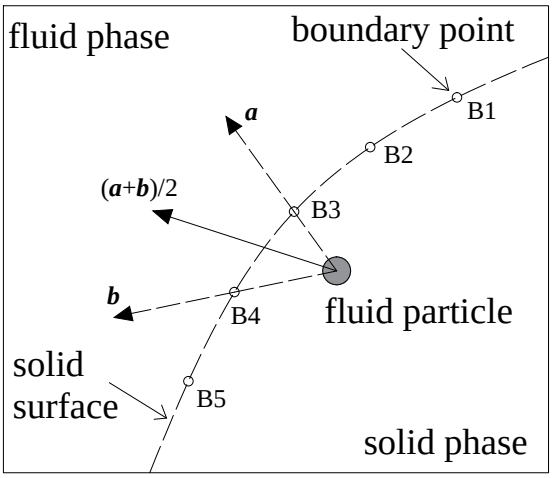


Figure 4-2: Schematic of the numerical technique designed to project fluid particles back into the fluid domain.

instead of using the boundary points, the centres of the triangular elements used to discretise the structure interface are checked to find the one that is closest to the fluid particle having entered the solid object, and the unit outward normal vector perpendicular to the face of that element is used to replace $(\boldsymbol{a}+\boldsymbol{b})/2$ in Equation 4.14.

4.3 Cell fraction θ occupied by solids

This section describes the technique used to approximate the cell fraction, θ, that is occupied by the solid. The cell fraction is required by the DLM method to represent the solid as described above. In particular, in Equation 4.10 and Equation 4.11, θ is crucial for the computation of the unique velocity of the solid. As mentioned above, following Ardekani et al. (2007), the cell fraction, θ, is calculated separately in cell volumes that surround different velocity values in the staggered grid; this is demonstrated in Figure 4-3. For the approximation of the cell fraction, θ, at each velocity cell, the velocity cell is first divided into 64 sub-cells, and then the centre of each sub-cell is checked to see if it is inside the solid phase or not. A sub-cell is considered as being fully occupied by the solid if its centre is located inside the solid phase. This can lead to errors at those sub-cells along the solid boundary. However, it is expected that the errors can cancel out, despite the fact that the

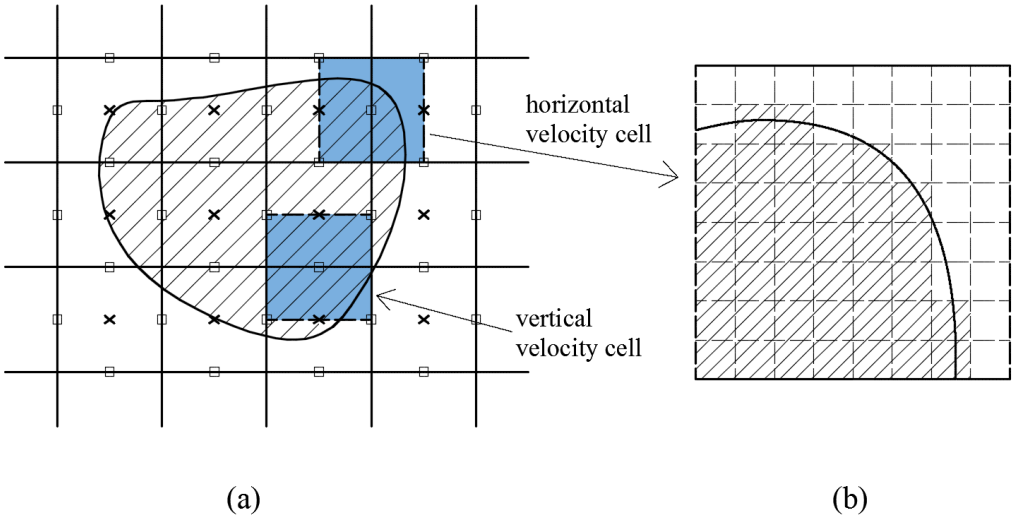


Figure 4-3: Schematic showing (a) the staggered grid and cell fraction occupied by the solid (slash line area). The shaded area represents the cell surrounding a velocity value, i.e. the velocity cell; (b) enlarged view of a horizontal velocity cell showing the detailed solid fraction (slash line area) computed in the numerical model.

errors themselves should already be very small as the size of the sub-cell is relatively small. An example of this implementation is visualised in Figure 4-3(b). The final value of θ is then taken to be the ratio of the total area of occupied sub-cells to the area of the full-sized containing cell.

4.4 Numerical algorithm

The solution procedure of the "full particle" PIC model, incorporating the DLM method for fluid-structure interaction, is summarised in Figure 4-4. As in chapter 3, the variables adjusted at each step of a computational cycle are also indicated.

4.5 Test cases

In this section, a test case of particle sedimentation is presented in which the present model is validated against experimental observations and numerical results from

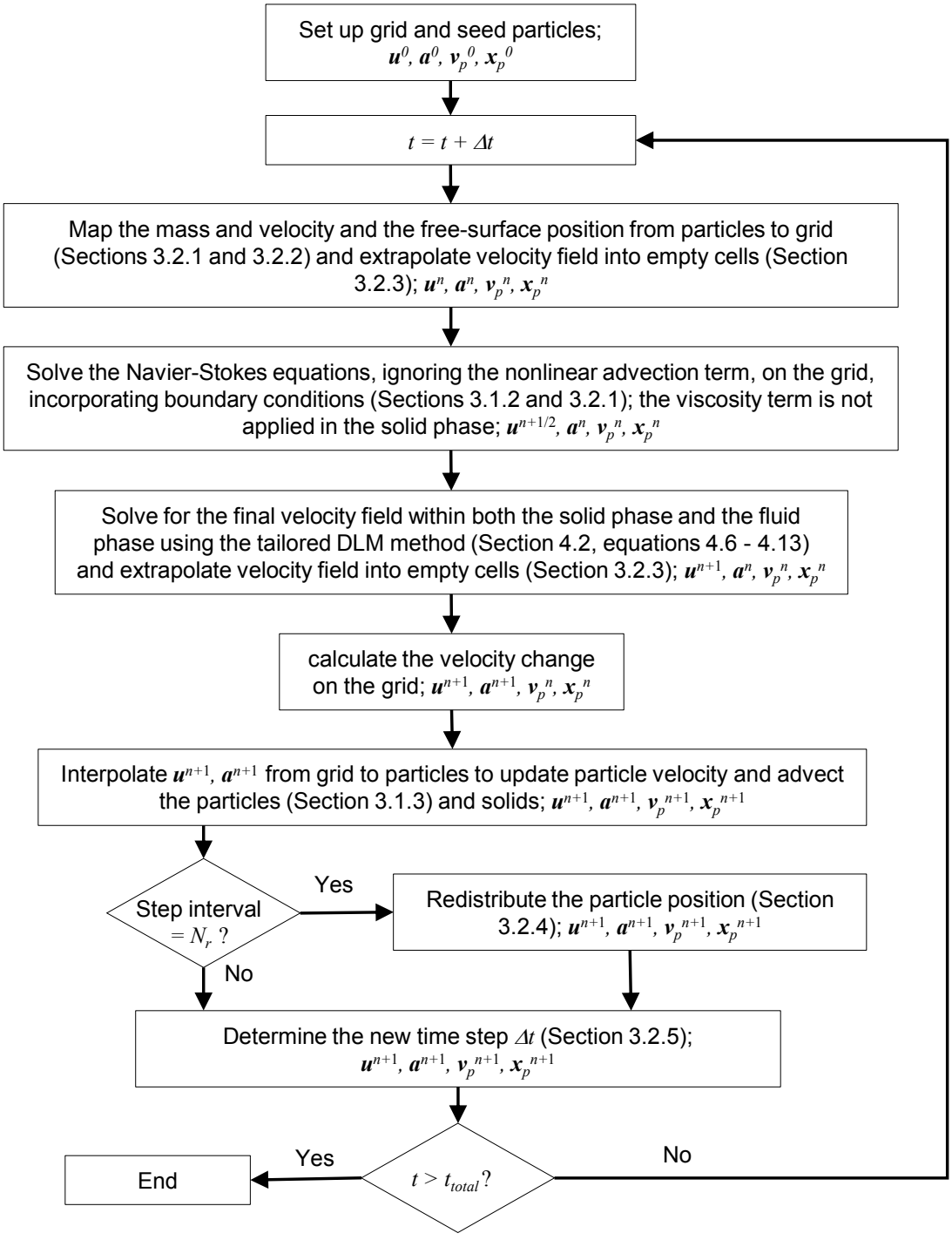


Figure 4-4: Flow chart of the solution procedure of the "full particle" PIC model, incorporating the DLM method for fluid-structure interaction.

other state-of-the-art models. Following that, the present model is applied to simulate a number of more realistic scenarios of rock dumping through vertical pipes. Attention has been focused on testing the model for its capability to capture the main physical process of fall-pipe rock dumping.

4.5.1 Sedimentation of two circular particles

The numerical model was first validated by modelling the sedimentation of two circular particles in a Newtonian flow. This benchmark test case was previously studied in Patankar et al. (2000) and Patankar (2001); similar test cases can be found in other numerical simulations (e.g. Glowinski et al. (1999)) and physical experiments (e.g. Fortes et al. (1987)).

Following Patankar (2001), the setup of the numerical model includes a 2D rectangular tank and two circular particles. The rectangular tank is 2 cm wide (x-direction) and 8 cm tall (z-direction) and is completely filled with water ($\nu = 1.0 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$, $\rho = 1.0 \times 10^3\,\mathrm{kg/m}^3$). The two circular particles have an identical radius of 0.1 cm and an identical density of $1.01 \times 10^3\,\mathrm{kg/m}^3$. At the time instant $t = 0$ s, the two circular particles are placed at the centre of the tank at a height of 6.8 cm and 7.2 cm, respectively. The particle placed at 6.8 cm height is hereafter referred to as the leading particle and the other one placed at 7.2 cm height is referred to as the lagging particle. The circular particles are allowed to fall freely due to gravity acting in the negative z-direction. Since the initial distance between the circular particles is small, according to Fortes et al. (1987), the motions of the particles would typically undergo three stages: drafting stage, kissing stage and tumbling stage. That is, after the circular particles are set to move, the lagging particle will be affected by the wake of the leading particle and thus tends to move faster (the drafting stage). Because of this, the two circular particles will quickly contact with each other along the sedimentation (the kissing stage). The system of the contacting particles is very unstable and a small offset of the centres of the particles will introduce an asymmetric wake that quickly separates the two circular particles (the tumbling stages).

In the present model, the collision force between the circular particles is simulated

using a formulation modified from that proposed in Glowinski et al. (1999):

$$\boldsymbol{F}_{i,j}^{p} = \begin{cases} \mathbf{0}, & d_{i,j} > R_i + R_j + \lambda, \\ \frac{1}{\epsilon_p} \frac{(\boldsymbol{r}_i - \boldsymbol{r}_j)(R_i + R_j + \lambda - d_{i,j})^2}{d_{i,j}}, & d_{i,j} \leqslant R_i + R_j + \lambda, \end{cases} \tag{4.15}$$

where $d_{i,j} = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|$, and $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the position vectors of the centres of the circular particles i and j, respectively; R_i and R_j are the radii of the circular particles i and j, respectively; λ is a distance range from which the circular particles start to exert force on each other; ϵ_p is a positive "stiffness" coefficient. Note that the value of the parameter ϵ_p is given below depending on the test cases under consideration, while λ, the distance range, is set to $2.0\Delta x$ consistently in this chapter. The parameter λ obviously causes unphysical buffer gaps between solids when they interact with each other. On the other hand, it can keep the solids separated and thus prevent one computational cell from being simultaneously occupied by different solids, which can lead to other numerical difficulties. Moreover, it can be expected that when the grid is sufficiently fine the influence of this parameter will become negligible.

For the numerical simulation, three sets of grid sizes were employed: $\Delta x = \Delta z =$ 0.01 cm (Case A), 0.016 cm (Case B) and 0.02 cm (Case C), resulting in 100×450 grid cells (with about 160,000 particles), 125×563 grid cells (with about 250,000 particles) and 200×900 grid cells (with about 640,000 particles), respectively. The parameter ϵ_p in Equation 4.15, as noted in Glowinski et al. (1999), is not critical to the computation of the contact force in this case and was set to the value of $1.0 \times 10^{-6}\Delta x$. The Courant number was set to 0.75. The CPU time for 6 s of simulation time on Case C, for example, was 56.5 min on an Intel Core i7-4600U (CPU @2.10Ghz∼2.70GHz) laptop with 8Gb RAM employing a Windows OS.

Figure 4-5 presents the snapshots of numerical simulation on Case C, showing the drafting, kissing and tumbling stages, respectively, which are qualitatively in agreement with the experimental observations of Fortes et al. (1987). The vorticity field shown in this figure demonstrates the fluid motion caused by the sedimentation of the two circular particles. Figure 4-6 further illustrates the fluid motion via the extent of vertical mixing of the fluid.

Figure 4-7 shows a grid refinement study on the vertical velocities of both circular

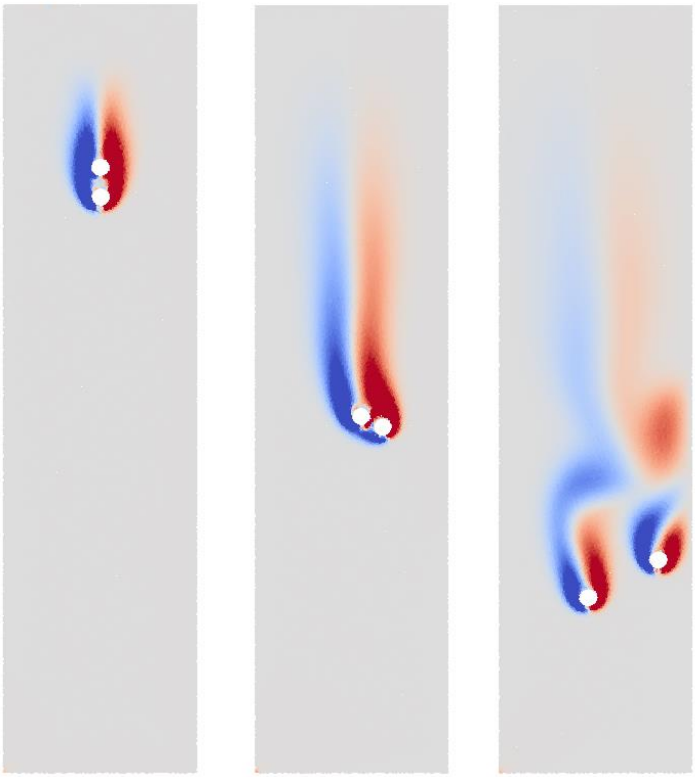


Figure 4-5: Snapshots of two circular particles sedimentation at $t = 1$ s, 3 s and 5 s (from left to right). The colour represents the contour of the dimensionless fluid vorticity $g^{-1/2}\nabla \times \boldsymbol{u}$, ranging from -1.5 (blue) to 1.5 (red).

particles for Cases A, B and C. It can be seen that the general trend of the velocities confirms the observations of the drafting, kissing and tumbling stages for all the Cases. The three Cases produce similar time periods for the drafting stage ($t = 0$–1.2 s), before the start of the kissing stage, but not for the start of the tumbling stage (e.g. $t = 1.5$ s for Case B). It is noted, as did Patankar (2001), that this discrepancy is inherent in these simulations, as the kissing and tumbling stages are essentially a manifestation of the instability of the falling particle system; the predictions rely heavily on the accuracy of the solutions of the numerical model and the simulation of solid collisions.

From Figure 4-7, it is also seen that when a coarse grid is used, the vertical velocities of both circular particles are smaller than those run by a fine grid at the ends of both the drafting and the tumbling stages. This may be due to the numerical dissipation induced by the use of Equation 4.10 and Equation 4.11 for the enforcement of the rigidity constraint as mentioned in Section 4.2, as well as the low order advection

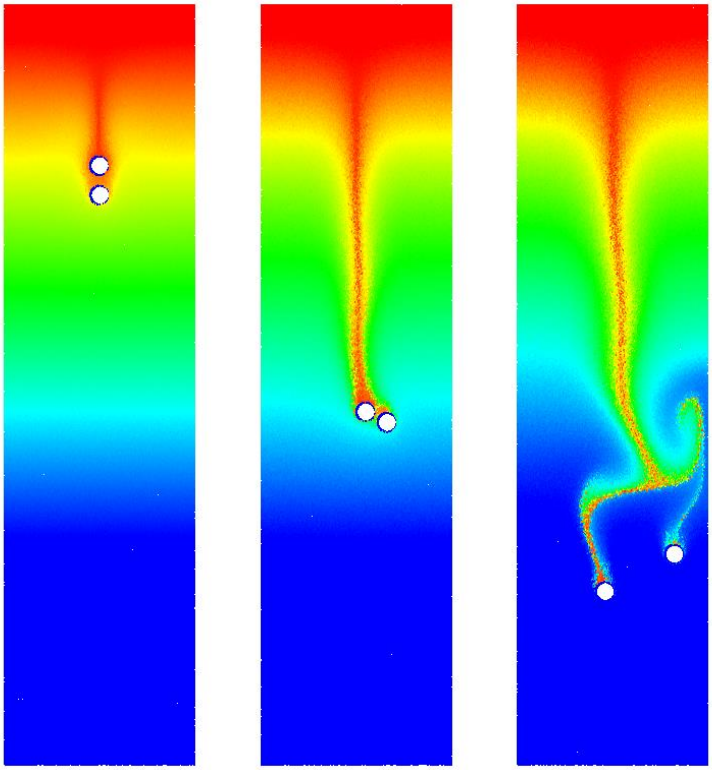
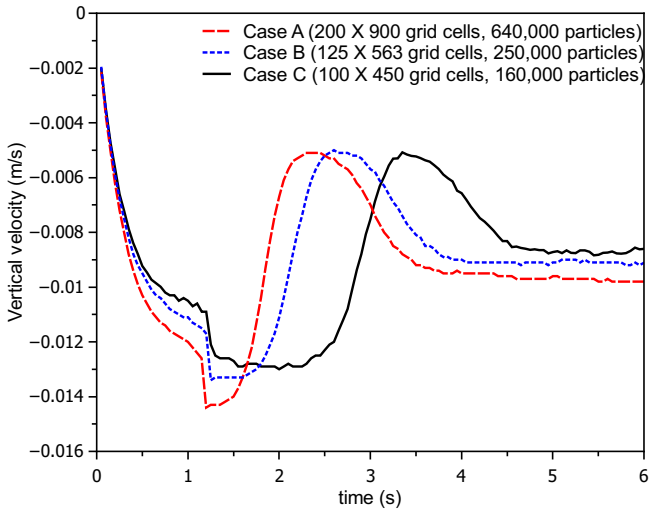


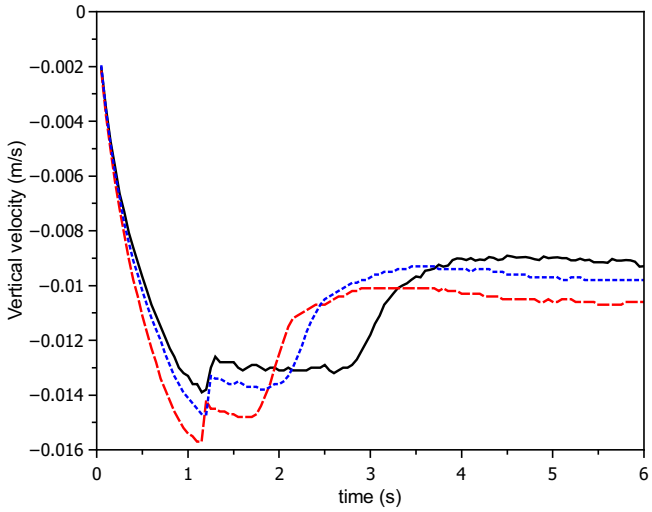
Figure 4-6: Snapshots of two circular particles sedimentation at $t = 1$ s, 3 s and 5 s (from left to right). The colour represents the vertical mixing extents.

scheme for the solids. It appears that the coarser the grid is, the less energetic the circular particles are. Similar results were also obtained when the Courant number was changed; the smaller Courant number led to more time-steps and hence more numerical energy dissipation. Nevertheless, as it can be seen, the overall discrepancy tends to become smaller when the grid is finer. Therefore, the results are convergent, considering the complexity of the problem. It should also be emphasised that the numerical model gives the same overall qualitative behaviour on the three different grid sizes. This is important to ensure the same macroscopic behaviour of the falling objects when a large number of solids are simulated (Patankar et al., 2000).

Figure 4-8 shows a comparison of the predicted vertical velocities of the two circular particles for the present model and the finite element based DLM model of Patankar (2001). For the present model, the results run on Case C are chosen for comparison, as the time step and the grid size for pressure in this case are both close to those used in Patankar (2001). From Figure 4-8, it can be seen that the results from the current model are quantitatively in good agreement with those from Patankar



(a) The leading particle



(b) The lagging particle

Figure 4-7: Comparisons of the time histories of vertical velocity of the circular particles run on three different grid sizes.

(2001), particularly before the start of the kissing stage, for which both models predict essentially the same time instant. Once the tumbling stage begins, the results from the two models start to show differences. Due to the highly stochastic nature of particle collisions, the results run by different models are unlikely to be identical, as even minor differences in the methodologies will be eventually amplified after the kissing stage.

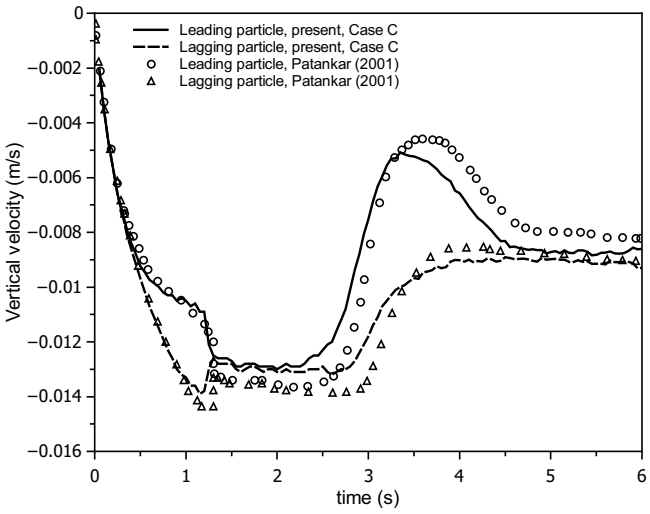


Figure 4-8: Comparison of the time history of vertical velocity for both leading and lagging particles.

4.5.2 Simulation of fall-pipe rock dumping

The collision force calculation

In this section, the numerical model is applied to simulate irregular rigid solids, resembling rocks, falling through vertical pipes filled with water. Because the shapes of the solids are irregular, Equation 4.15, which is designed for circular objects or spheres cannot be used directly. Here, the collision between solids is handled using the spring and dash-pot system similar to the discrete element method (DEM) proposed in Cundall and Strack (1979), making use of the boundary points on the solid surface (see Figure 4-1). The idea is to first compute the contact force, $\boldsymbol{f}_{ci}$, exerted on each individual boundary point by the boundary point of other solids and/or the domain walls, and then sum up all the forces on the individual boundary points of each solid to obtain an overall force $\boldsymbol{f}_{co}$ and momentum M_{co}:

$$\boldsymbol{f}_{co} = \sum_{i}^{N_b} \boldsymbol{f}_{ci}\,; \tag{4.16}$$

$$M_{co} = \sum_{i}^{N_b} \boldsymbol{f}_{ci} \times \boldsymbol{r}_{bpi}\,, \tag{4.17}$$

where N_b represents the number of boundary points of a solid and $\boldsymbol{r}_{bpi}$ is a position

vector pointing from the rotation centre of the solid to its ith boundary point. The contact force, $\boldsymbol{f}_{ci}$, on the boundary point i is computed by:

$$\boldsymbol{f}_{ci} = \boldsymbol{F}^{p}_{i,j} + \boldsymbol{F}^{w}_{i,j} + \boldsymbol{F}^{d}_{i,j}\,, \tag{4.18}$$

where the subscript j represents the index of the boundary point of another solid that is closest to the boundary point i, or the index of an imaginary mirror of the boundary point i with respect to a wall boundary; $\boldsymbol{F}^{p}_{i,j}$ is computed by Equation 4.15 and accounts for the contact force between the boundary points of different solids, while $\boldsymbol{F}^{w}_{i,j}$, which accounts for the contact force between the boundary point and a wall boundary, is computed following Glowinski et al. (1999):

$$\boldsymbol{F}^{w}_{i,j} = \begin{cases} \boldsymbol{0}, & d_{i,j} > \lambda, \\ \frac{1}{\epsilon_w}\frac{(\boldsymbol{r}_i - \boldsymbol{r}_j)(\lambda - 0.5d_{i,j})^2}{d_{i,j}}, & d_{i,j} \leqslant \lambda, \end{cases} \tag{4.19}$$

where $d_{i,j} = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|$, and $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the position vectors of the boundary point i and its imaginary mirror j of a wall boundary; ϵ_w is a positive "stiffness" parameter whose value is given below depending on the test case under consideration; λ is the same distance range as that used in Equation 4.15; $\boldsymbol{F}^{d}_{i,j}$ is an additional damping force whenever $\boldsymbol{F}^{p}_{i,j}$ or $\boldsymbol{F}^{w}_{i,j}$ is nonzero and is computed by:

$$\boldsymbol{F}^{d}_{i,j} = -\left(\epsilon_d(\boldsymbol{v}_i - \boldsymbol{v}_j)\cdot\frac{\boldsymbol{r}_{ij}}{\|\boldsymbol{r}_{ij}\|}\right)\frac{\boldsymbol{r}_{ij}}{\|\boldsymbol{r}_{ij}\|}\,, \tag{4.20}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ is the position vector; $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the velocities at the boundary point i and point j, respectively; ϵ_d is a stiffness parameter and was set to the value of $5.0 \times 10^5 \Delta x$ in the following simulations. It is noteworthy that, for numerical stability, when the relative velocity $\boldsymbol{v}_i - \boldsymbol{v}_j$ is positive (i.e. the two points are moving away from each other), the absolute value of $\boldsymbol{F}^{d}_{i,j}$ is restricted to be no larger than the values of either $\boldsymbol{F}^{p}_{i,j}$ or $\boldsymbol{F}^{w}_{i,j}$ that is associated. This prevents two departing points from being pulled back by the damping force and overlapping with each other, when the relative velocity is quite high. Also, in the present implementation of Equation 4.20, when j represents an imaginary point of the boundary point i, $\boldsymbol{v}_j$ is set to zero. Furthermore, it is noted that the boundary points are not augmented with a radius. Thus, the radius, R, in Equation 4.15 is omitted when the equation is used here. Moreover, the parameters ϵ_p and ϵ_w in Equation 4.15 and Equation 4.19 were both set to the value of $1.0 \times 10^{-6} \Delta x$, via numerical experiments for stability.

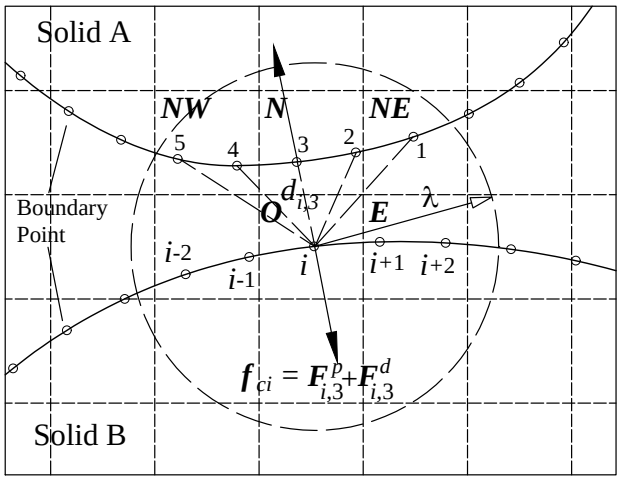


Figure 4-9: Schematic showing the contact force for the boundary point i when two solids interact with each other. $d_{i,3} = \min(d_{i,1}, d_{i,2}, d_{i,3}, d_{i,4}, d_{i,5}) \leq \lambda$.

Figure 4-9 visualises an example of the contact force computation for boundary point i of solid B when two solids A and B interact with each other. For simplicity of presentation, while the boundary points of solid B are indexed using letters, those of solid A are indexed using numbers. In the numerical implementation, when the boundary point i is identified, a search in grid cells $\boldsymbol{O}, \boldsymbol{E}, \boldsymbol{NE}, \boldsymbol{N}$ and $\boldsymbol{NW}$ is conducted to find out the boundary point that belongs to another solid object and is closest to the boundary point i; as shown in Figure 4-9, the boundary point 3 of solid A is the one that meets these criteria. Once this is done, the contact force $\boldsymbol{f}_{ci}$ on the boundary point i of solid B is calculated using Equation 4.18, based on the boundary point 3 of solid A. Meanwhile, $-\boldsymbol{f}_{ci}$ is added to the contact force $\boldsymbol{f}_{c3}$ on the boundary point 3 of solid A. This is because in the current numerical implementation, the boundary point 3 of solid A may be also chosen to calculate the contact force by other boundary points of solid B.

When collision between solids occurs, the solid position is updated following Patankar et al. (2000), combining both the solid velocity and the collision force. The updated position of a solid due to, for example, the translational movement, is computed by the following procedure:

(1) Set $\boldsymbol{Y}^{n+1,0} = \boldsymbol{Y}^{n}$ and $\boldsymbol{f}_{co}(\boldsymbol{Y}^{n+1,0}_{\text{temp}}) = \boldsymbol{f}_{co}(\boldsymbol{Y}^{n})$;
(2) do $k = 1, K$

$$\boldsymbol{Y}^{n+1,k}_{\text{temp}} = \boldsymbol{Y}^{n+1,k-1} + \left(\frac{\boldsymbol{U}^{n} + \boldsymbol{U}^{n-1}}{2}\right)\left(\frac{\Delta t}{K}\right) \tag{4.21}$$

$$\boldsymbol{Y}^{n+1,k} = \boldsymbol{Y}^{n+1,k}_{\text{temp}} + \frac{1}{2}\left(\frac{\boldsymbol{f}_{co}(\boldsymbol{Y}^{n+1,k-1}_{\text{temp}}) + \boldsymbol{f}_{co}(\boldsymbol{Y}^{n+1,k}_{\text{temp}})}{2M}\right)\left(\frac{\Delta t}{K}\right)^{2} \tag{4.22}$$

enddo;
(3) Set $\boldsymbol{Y}^{n+1} = \boldsymbol{Y}^{n+1,K}$.

In which, $\boldsymbol{Y}$ is the position vector of a solid, $\boldsymbol{U}$ is the solid translational velocity vector (Equation 4.10) and Δt is the numerical time step used in the fluid computation. The number of sub-steps, K, is chosen to be 50 to stabilise the numerical model when dealing with the solid collisions. Once the position of the solid is updated according to the above translational movement, the translational acceleration $\boldsymbol{A}_c$ of the solid (see Equation 4.9) is computed as:

$$\boldsymbol{A}^{n+1}_{c} = \frac{2}{(\Delta t)^2}\left(\boldsymbol{Y}^{n+1} - \boldsymbol{Y}^{n} - \left(\frac{\boldsymbol{U}^{n} + \boldsymbol{U}^{n-1}}{2}\right)\Delta t\right) \ . \tag{4.23}$$

Note that the solid position due to the rotational movement (due to the momentum M_{co} calculated by Equation 4.17) and the angular acceleration $\boldsymbol{\alpha}_c$ in Equation 4.9 are computed in the same manner.

It should be noted that, due to the assumptions and selections of parameters given above, the method described in this section may not be able to model rock collisions very accurately in realistic scenarios, for which further developments could refer to the methods proposed in Cundall and Strack (1979) and Gotoh et al. (2009).

Test case: fall-pipe rock dumping in 5 m water depth

This test case concerns 46 rocks dumped through a continuous vertical pipe. Figure 4-10 shows the dimensions of the numerical setup. A tank of 7.0 m high and 9.0 m wide is filled with water of 5.0 m in depth. A pipe with a funnel on the top is inserted into water at the middle of the tank. The pipe has an inner width of 0.96 m and is fixed at 1.5 m above the tank bottom. The rocks of irregular shapes are initially placed in the funnel and are allowed to fall down through the pipe at $t =$

0 s. The rocks have a dimension of roughly 0.3 m $\times$ 0.3 m, and a density, ρ_S, of $2.5 \times 10^3\,\mathrm{kg/m^3}$.

For the numerical simulation, the grid size was $\Delta x = \Delta z = 0.02$ m, resulting in 350×450 cells with approximately 467,000 particles. The distance between two successive boundary points for all rocks was approximately $0.15\Delta x$. The Courant number was 0.75, and the particle redistribution scheme (see Section 3.2.4) was used every $N_r = 15$ time-steps to ensure a smoother particle field. The rigid pipe, as well as the funnel, was simulated by applying the same boundary conditions as those of the domain boundary. Also, the shapes of the rocks are characterised as being round (Latham et al., 2008). Thus, the contact area between the rocks is assumed to be small and the friction force in the tangential direction is omitted. However, in order to stop the rocks from sliding along the bed of the tank when they fall onto it, a friction force $\boldsymbol{F}_i^f$ is added to the contact force between the boundary point i and its imaginary point with respect to the bed of the tank: $\|\boldsymbol{F}_i^f\| = 0.65\|\boldsymbol{F}_{i,j}^w\|$, where $\boldsymbol{F}_{i,j}^w$ is calculated using Equation 4.19 and the direction of the friction force $\boldsymbol{F}_i^f$ is the opposite of that of the horizontal velocity component at the boundary point i. For this case, it took about 15.6 h for 20 s of simulation on an Intel Core i7-4930K (CPU

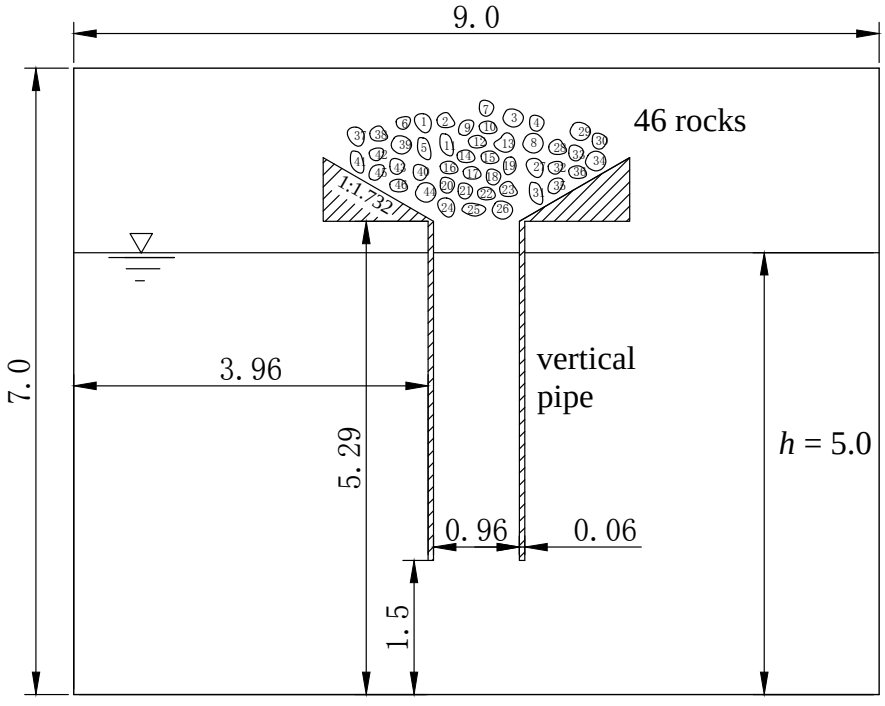


Figure 4-10: Numerical setup for rock dumping through a vertical pipe; unit in (m).

@3.40Ghz) desktop with 32Gb RAM employing a Windows OS.

Figure 4-11 presents snapshots of the numerical simulation. It is seen from Figure 4-11(a)–(d) that as the rocks fall down into the pipe, the inner free-surface level decreases, while the outer free-surface level increases. This, in turn, leads to a reverse flux inside the pipe, due to the imbalance of pressure inside and outside the pipe. This phenomenon was found to continue as the free surface oscillates, resulting in a discontinuous fall of the rock and eventually a delay of the rock dumping. For this case, it takes more than 16 s to dump all the rocks through the pipe.

To double check the effect of the pipe, another simulation was conducted in which the fall-pipe was not used. Snapshots of results from this test are presented in Figure 4-12, from which it is clearly seen that, without the pipe, the rocks tend to scatter as they fall through the water. This dispersion of rocks may be a result of not only the asymmetric wake (and hence fluid forces) caused by the rocks of irregular shapes themselves as they fall down, but also the drafting, kissing and tumbling effects mentioned above. The final distributions of rocks on the bed, as shown in Figure 4-12(b) and Figure 4-11(f), confirm that the fall-pipe can effectively restrict the dumping area of rocks.

The third simulation concerns a perforated pipe. The pipe presented in Figure 4-10 was modified by inserting 0.1 m diameter holes symmetrically on both sides every 1.0 m from the bottom upwards. The snapshots of this simulation are shown in Figure 4-13. It can be seen that the rocks are dumped out very quickly; the dumping time (approximately 6 s) is reduced by more than a half, compared with that of the continuous pipe case. The primary reason for this is that the reverse flux did not grow inside the pipe as the existence of holes contributed to the quick equalisation of the free-surface level (hence the pressure) at both sides of the pipe. In fact, the drag induced by the rocks creates a downward flow in the pipe, which in turn enhances the falling speed of the rocks. This is typical of the behaviour resulting from semi-open fall-pipes (Beemsterboer, 2013).

Test case: fall-pipe rock dumping in 10 m water depth

In order to test the current model for simulating a longer pipe with a larger water depth, the pipe is extended to 11 m long, with a water depth of 10 m. For this

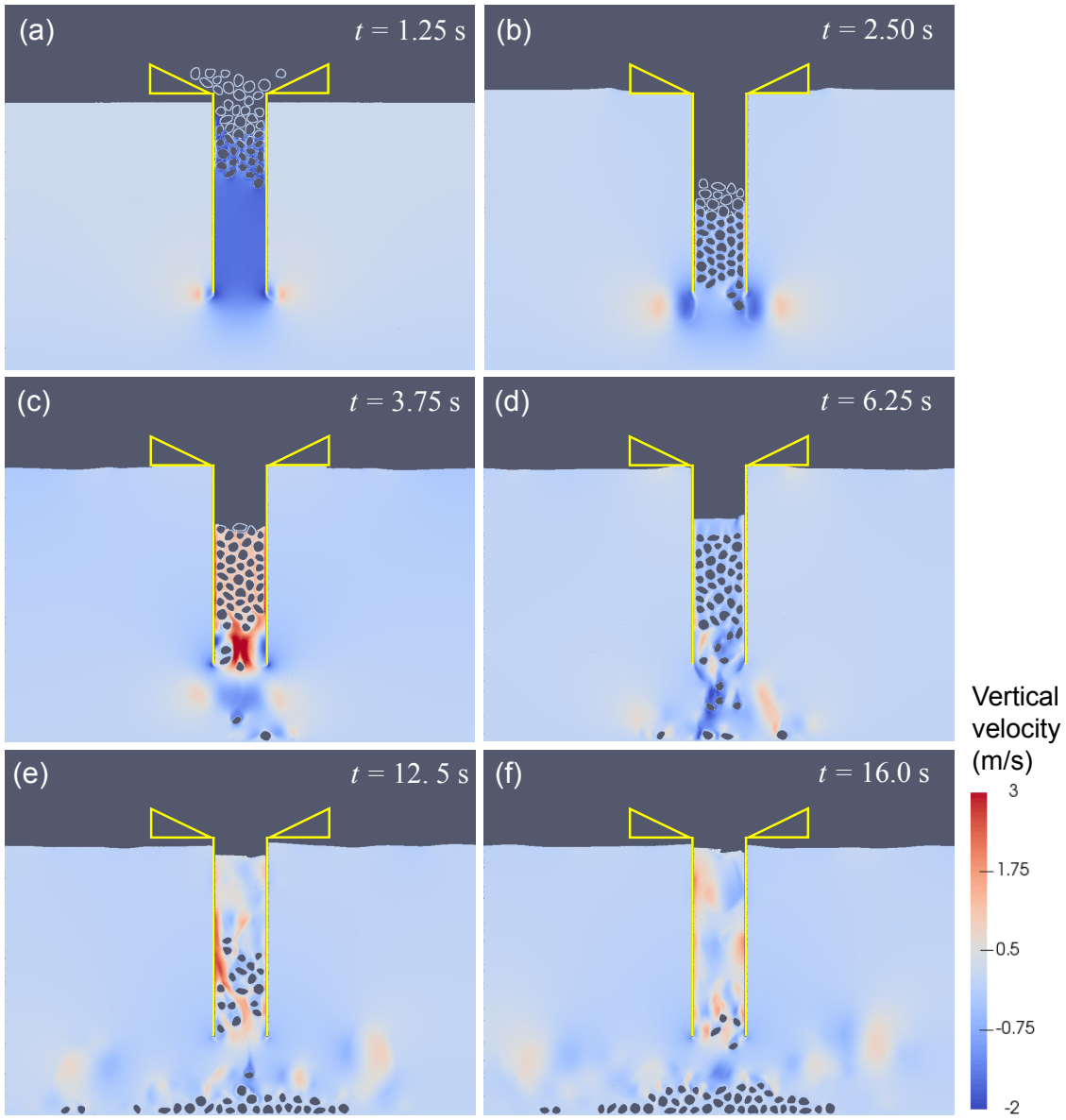


Figure 4-11: Snapshots of numerical simulation for 46 rocks dumped through a continuous vertical pipe.

case, 52 rocks are initially placed just under the still free surface, and the bottom of the pipe is sealed. A schematic of the setup of this case is shown in Figure 4-14(a). Other numerical settings are the same as those used in the 5 m water depth case, resulting to a grid resolution of 50×550 with approximately 106,000 particles. This case took around 5.5 h for 20 s of simulation on the same desktop machine.

Figure 4-14(b)–(g) provide snapshots of the numerical simulation. It is seen that,

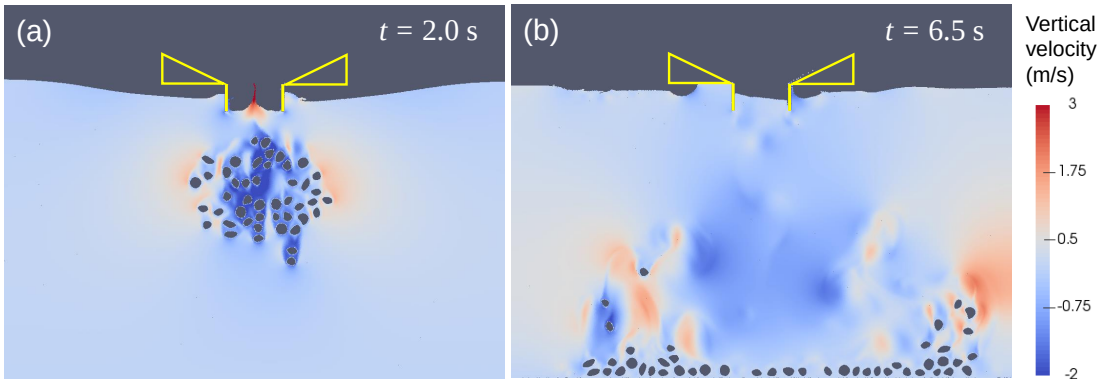


Figure 4-12: Snapshots of numerical simulation for 46 rocks dumped through a short vertical pipe.

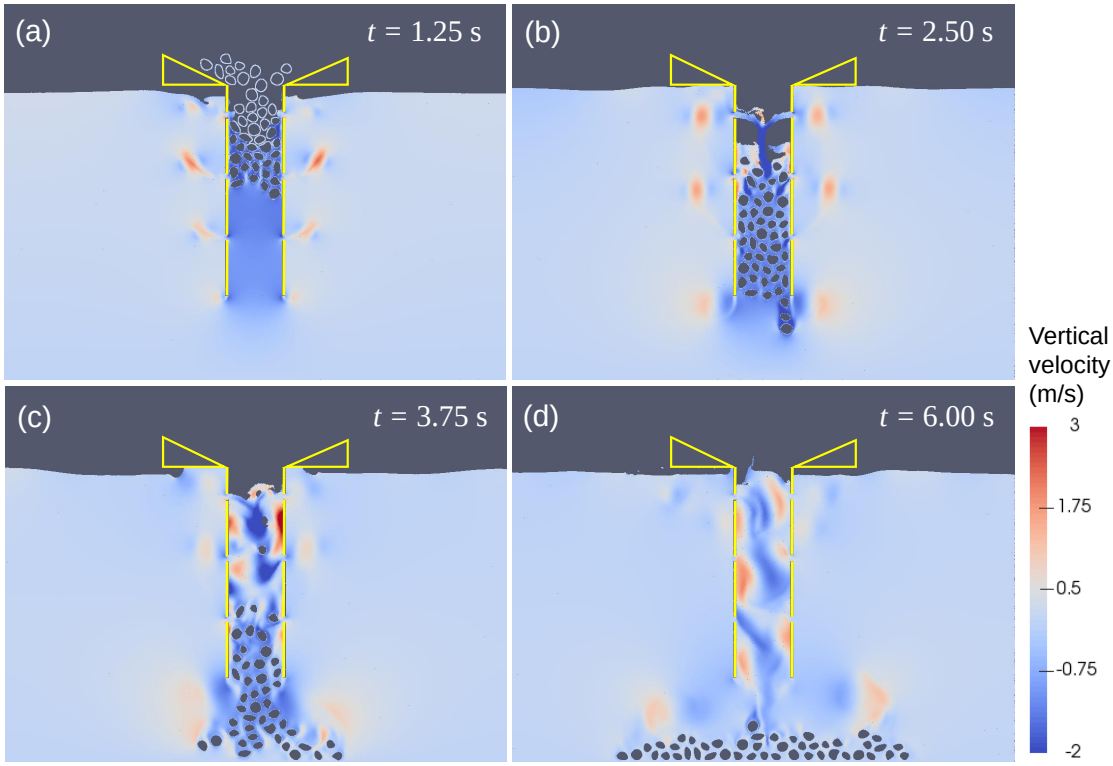


Figure 4-13: Snapshots of numerical simulation for 46 rocks dumped through a perforated vertical pipe.

as the rocks fall down through the pipe, upward fluid fluxes are generated at the same time. This is because the bottom of the pipe is closed. These fluxes separate the rocks and when they reach the free surface, large free-surface deformations are observed (see e.g. Figure 4-14(f)). It is also seen that as the rocks initially located at the lower layers quickly fall and hit the bottom, the other rocks initially placed at

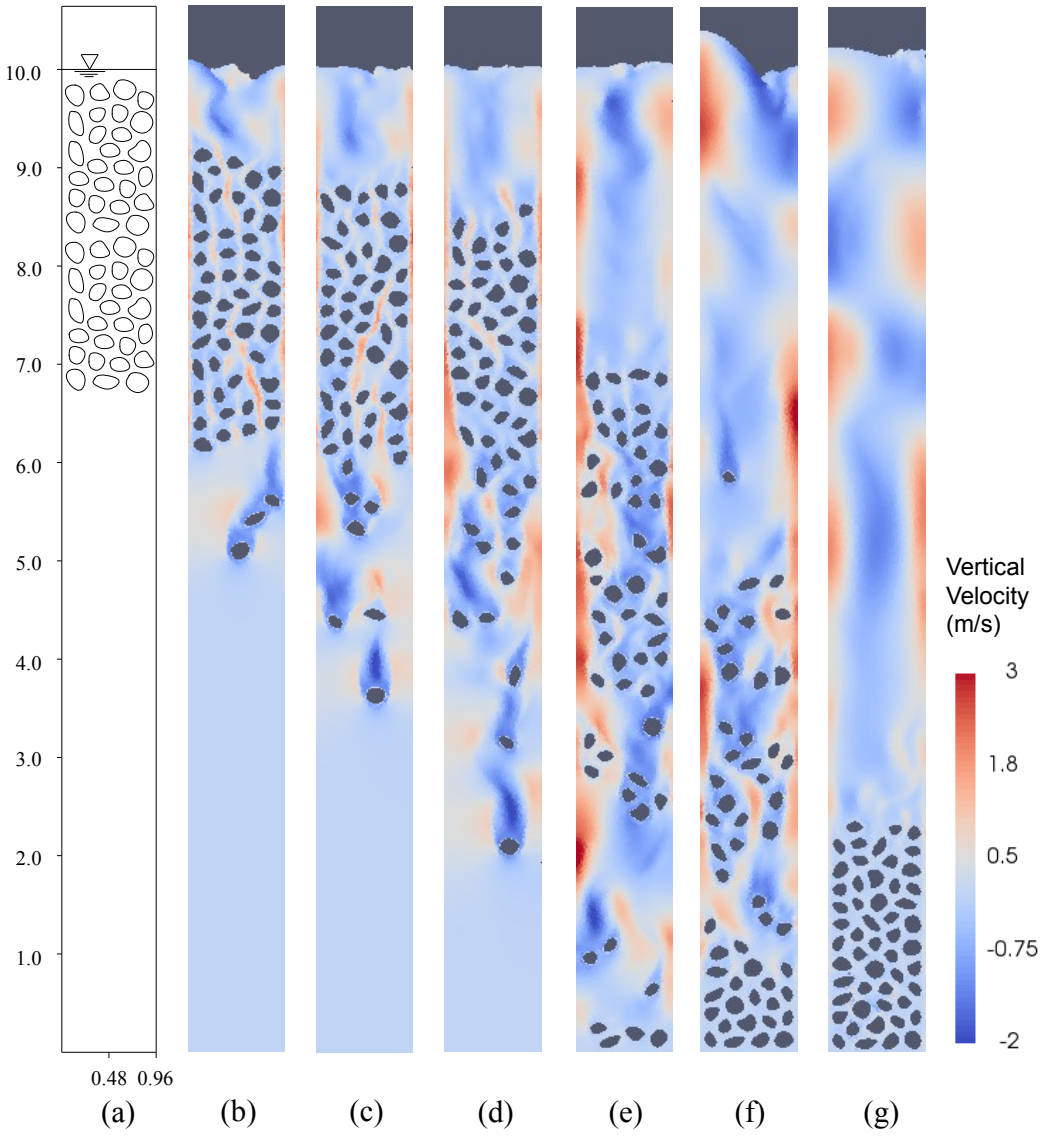


Figure 4-14: Snapshots of numerical simulation for 52 rocks dumped through a pipe, at $t = 0.0$ s, 2.5 s, 3.75 s, 5.0 s, 10.0 s, 15.0 s and 20.0 s from (a) to (g), respectively. The dimensions in (a) are in metre.

the middle and upper layers are delayed. This is because the rocks at the middle and upper layers encountered more upward fluxes (and hence a larger fluid resistance) that are caused by the movement of the rocks at the lower layer. This could lead to rock clustering (even clogging) in pipes in realistic scenarios.

4.6 Summary

In this chapter, the "full particle" PIC framework developed in chapter 3 was augmented by a tailored version of the DLM approach proposed in Patankar (2001) for fluid-solid interaction. The solids in this scheme are first solved as if they were fluids, and then a correction step is utilised to modify the velocity inside the solid phase due to density differences between fluids and solids, and solid collisions. Finally, the rigidity constraint is imposed through enforcing the mass and momentum conservation laws.

The current model was validated against a benchmark test of two circular particles falling down in a 2D rectangular tank; the results are qualitatively in agreement with the experimental observations, and quantitatively match those from other state-of-the-art numerical models. The model was then applied to the simulation of fall-pipe rock dumping. It seems that the model is able to capture the key elements of this process such as the effects of the fall-pipe on productivity and the potential initiation of rock clustering in pipes. It is expected that the current model could also be useful in simulating other, similar problems such as armor blocks in front of caisson breakwaters (Gotoh et al., 2009) and rock armor around wind turbine foundations (Whitehouse et al., 2011).

It has been shown that the current DLM model is straightforward to implement for the modelling of interactions between fluid and a large number of solids. On the other hand, the algorithm used in this chapter may have difficulty in ensuring a divergence-free velocity field in cells shared by solid and fluid, as well as conserving the kinetic energy of moving solids. In addition, because the current DLM method is designed for freely moving solids, further development is required if this method is to be used to simulate prescribed motion or limited degree of freedoms (DoF) structures such as wavemakers.

CHAPTER 5

A CARTESIAN CUT CELL BASED TWO-WAY STRONG FLUID-STRUCTURE COUPLING ALGORITHM FOR WAVE-STRUCTURE INTERACTION

In this chapter, a Cartesian cut cell based two-way strong fluid-structure coupling algorithm is incorporated in the "full particle" PIC framework developed in chapter 3. Attention has been focused on simulating water wave interaction with coastal and offshore structures in two spatial dimensions. This involves different types of water waves interacting with structures of different shapes and degree of freedoms (DoF). To investigate these topics, a numerical wave tank (NWT) has been established in the PIC model. Results of test cases show that the modified model is able to both efficiently and robustly predict violent wave motions, wave forces on structures and structure motions under different wave action.

5.1 Introduction

In a coupled fluid-solid system, when the influence from one aspect to another becomes negligible, a one-way coupling will work well, e.g. the simulation of a piston-type wavemaker. In most cases, however, both the solid and the fluid phases interact with each other and a full two-way fluid-solid coupling scheme is required, see e.g. Hadžić et al. (2005), Batty et al. (2007) and Zhao et al. (2014). Among these schemes, when solid structures are only used to provide velocity boundaries for the solution in the fluid domain and the fluid is solely employed to give a pressure boundary for solid computations, they are usually termed weak coupling (see e.g. Hadžić et al. (2005) and Zhao et al. (2014)). In this sense, strong coupling is defined to mean that

the effects of fluid and solid boundaries are treated simultaneously. Typical examples of strong coupling schemes can be found in Patankar et al. (2000) and Batty et al. (2007). The current "full particle" PIC model is only suitable for incompressible fluids. Within this framework, it was found that when implementing a weak coupling scheme, especially for simulating floating structures, the behaviour of pressure in the fluid domain can become very stiff which causes numerical instabilities in the calculations of both the fluid and solid motions. The main reason for this is that in the present solver, the fluid pressure is used not only to update the solid domain but also to maintain the flow incompressibility; therefore, an inaccurate value for the velocity change on the solid boundaries will cause large pressure fluctuations inside the fluid phase, which in turn exerts an incorrect force on the structure and can ultimately cause the numerical model to fail. This type of stiff pressure behaviour in algorithms for incompressible flow has been discussed previously by Fedkiw (2002). Although this unstable pressure behaviour can potentially be alleviated by iterating the solver to a convergent state (see e.g. Hadžić et al. (2005) and Borazjani et al. (2008)), the CPU cost is, in many cases, prohibitively expensive. Thus, strong fluid-solid coupling schemes are the preferred approach to be adopted in this thesis.

In chapter 4, the model uses a modified version of the DLM method of Patankar (2001) for fluid-solid interactions. This method is efficient and capable of simulating interactions between multiple solids and fluid. However, since in the DLM approach the solids are treated as fluids in the initial stages, when simulating floating structures, the surface-piercing portion of the structure needs to be treated as the free surface of the fluid. It is necessary, and non-trivial, to consistently apply the boundary conditions, especially in the presence of angled corners such as floating boxes. The previous DLM approach thus requires further improvement and validation within the current framework when it is to be used for simulating surface-piercing floating structures for engineering applications. For this reason, this chapter instead focuses on developing a robust and efficient strong coupling approach based on the cut cell technique of Ng et al. (2009) that is modified according to the strong coupling idea presented by Batty et al. (2007). Details of this approach are described in the following sections.

5.2 Governing equations

The fluid part is simulated by solving the incompressible Newtonian NS equations:

$$\nabla \cdot \boldsymbol{u} \;=\; 0, \tag{5.1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \boldsymbol{f} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u}, \tag{5.2}$$

where $\boldsymbol{u}$ is the velocity; t is time; $\boldsymbol{f}$ is the external force; p is the fluid pressure; ν is the liquid kinematic viscosity. The numerical approaches are described in chapter 3.

5.3 Numerical implementations of the fluid-structure coupling algorithm

5.3.1 Boundary conditions

Figure 5-1 shows a typical setup of the 2D NWT, where different types of boundaries are depicted. For the free surface, ζ, and the domain boundary, $\partial\Omega$, the boundary conditions applied are described in chapter 3. For the surfaces of structures, $\partial\Omega_S$, the boundary conditions enforced are:

$$\boldsymbol{n} \cdot \boldsymbol{u} = \boldsymbol{n} \cdot \boldsymbol{U}_b \quad \text{and} \quad \boldsymbol{n} \cdot (\Delta t \rho^{-1} \nabla p) = \boldsymbol{n} \cdot (\tilde{\boldsymbol{U}}_b - \boldsymbol{U}_b^{n+1}) \quad \text{on } \partial\Omega_S(\boldsymbol{x}, t), \tag{5.3}$$

where $\boldsymbol{U}_b$ is the velocity field imposed on the structure boundary, with the superscript $n+1$ representing the time-step; $\boldsymbol{n}$ is the unit outward normal vector; $\tilde{\boldsymbol{U}}_b$ represents the tentative boundary velocity before the pressure projection step. These boundary conditions will be resolved within the discretisation of the PPE; this is described in the following section.

Handling solid boundaries, including those of wavemaker and other structures as shown in Figure 5-1, differs in a few details. The viscous term is solved explicitly on the grid by applying a free-slip condition to the wavemaker boundary whilst a no-slip condition is implemented on the boundaries of other structures. Here, the

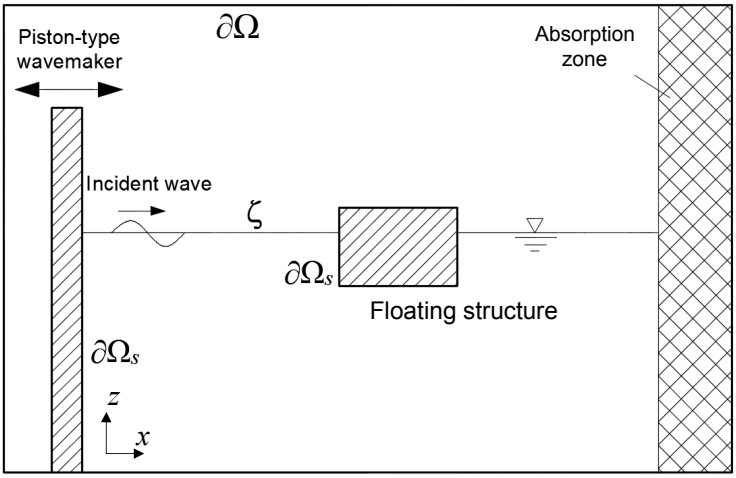


Figure 5-1: Schematic showing the setup of a 2D NWT.

no-slip condition is approximately enforced by imposing the structure velocity on the solid phase; that is, the velocity components of cell edges (cell faces in three spatial dimensions) that are fully occupied by structures are calculated based on the overall velocity of the structure. The advection term is solved via particles in a Lagrangian manner, and this involves interpolating velocities from the grid. Note that when modelling a fixed structure, the velocities at grid faces that are fully occupied by the structure are simply zero. Thus, interpolating velocities for displacing particles around the structure boundaries may cause them to "stick onto" the structure.

To mitigate this effect, an extrapolation of the fluid velocity into the solid phase was implemented for fixed structures, using the same technique as the one described in Section 3.2.3 for the free-surface. Note that, here, a SDF, ϕ_s, with respect to the structure interface is constructed at the grid vertices using the same method as that described in Section 3.2.1. In particular, the structure interface is represented by a set of boundary points (see Figure 5-2), which are used to compute the seed values of ϕ_s around the structure. It is noted that during one computational step the velocity extrapolation is conducted before and after the solution of the Navier-Stokes equations for the purpose of calculating the velocity change in the atmospheric pressure region and regions occupied by fixed structures (see Figure 5-5), which is required by the WENO interpolation scheme as four-point stencils are required (see Section 3.2.2).

It is worth noting that the fluid velocities are also extrapolated into the wavemaker

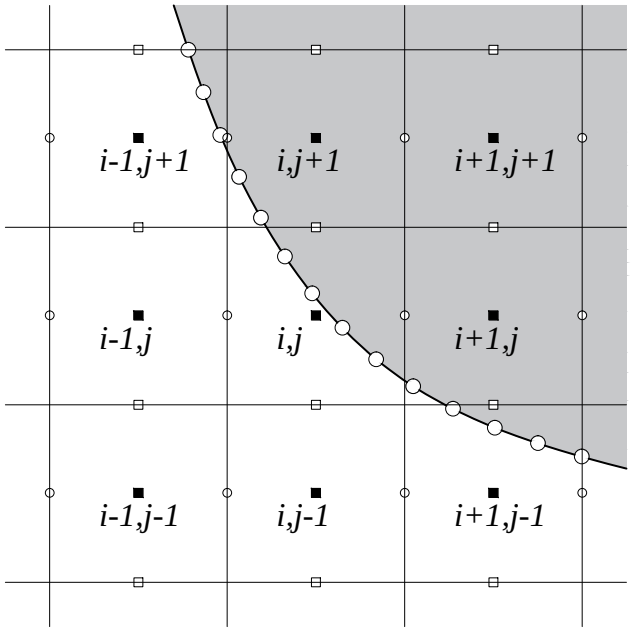


Figure 5-2: The staggered grid and a structure (the shaded area). The boundary of the structure is discretised into a set of equally spaced boundary points.

phase in a slightly different manner. Since the fluid-structure interface of the piston-type wavemaker always vertically cuts the grid cells, in this study, the extrapolated velocities inside the wavemaker for the z-direction (and also y-direction in three spatial dimensions) are simply copied from the adjacent fluid cells, while velocities for the x-direction within the wavemaker phase are determined according to the wave generation techniques described in Section 5.5. However, it is noted that the velocity change inside the wavemaker phase in all directions are copied from the adjacent fluid cells, after the velocity change in the fluid phase has been calculated.

5.3.2 Finite volume discretisation of PPE on irregular domains

The Cartesian cut cell based two-way strong fluid-structure coupling algorithm is described in this section. This involves solving the PPE on irregular domains cut by structures, using a finite volume discretisation. The derivations here are presented in two spatial dimensions; extension to three spatial dimensions is straightforward.

The algorithm is based on the cut-cell type approach used in Ng et al. (2009). Let Ω_f be the fluid phase and G_{ij} represent cell (i, j). By applying the finite volume

approach to the PPE (Equation 3.7) and evoking the divergence theorem, we have

$$\Delta t \rho^{-1} \int_{\partial(G_{ij} \cap \Omega_f)} \boldsymbol{n} \cdot \nabla p^{n+1} \, dA = \int_{\partial(G_{ij} \cap \Omega_f)} \boldsymbol{n} \cdot \tilde{\boldsymbol{u}} \, dl, \tag{5.4}$$

where dl and dA denote the length and the area differentials in two spatial dimensions, respectively; $\boldsymbol{n}$ is the unit normal vector pointing out of the solid. Rewriting the boundary integral on both the cell edges and the solid interface in Equation 5.4 into its approximation form gives:

$$\begin{aligned}\int_{\partial(G_{ij} \cap \Omega_f)} \boldsymbol{n} \cdot \nabla p^{n+1} \, dA \simeq E_{i-\frac{1}{2},j} \cdot \frac{(p^{n+1}_{i-1,j} - p^{n+1}_{i,j})}{\Delta x} + E_{i+\frac{1}{2},j} \cdot \frac{(p^{n+1}_{i+1,j} - p^{n+1}_{i,j})}{\Delta x} + \\ E_{i,j-\frac{1}{2}} \cdot \frac{(p^{n+1}_{i,j-1} - p^{n+1}_{i,j})}{\Delta z} + E_{i,j+\frac{1}{2}} \cdot \frac{(p^{n+1}_{i,j+1} - p^{n+1}_{i,j})}{\Delta z} - \int_{G_{ij} \cap \partial\Omega_S} \boldsymbol{n} \cdot \nabla p^{n+1} \, dA\end{aligned} \tag{5.5}$$

and likewise:

$$\begin{aligned}\int_{\partial(G_{ij} \cap \Omega_f)} \boldsymbol{n} \cdot \tilde{\boldsymbol{u}} \, dl \simeq E_{i+\frac{1}{2},j} \cdot \tilde{u}_{i+\frac{1}{2},j} - E_{i-\frac{1}{2},j} \cdot \tilde{u}_{i-\frac{1}{2},j} + E_{i,j+\frac{1}{2}} \cdot \tilde{w}_{i,j+\frac{1}{2}} - E_{i,j-\frac{1}{2}} \cdot \tilde{w}_{i,j-\frac{1}{2}} \\ - \int_{G_{ij} \bigcap \partial\Omega_S} \boldsymbol{n} \cdot \tilde{\boldsymbol{U}}_b \, dl,\end{aligned} \tag{5.6}$$

where E is a fraction of grid face that is open to water, i.e. not occupied by solid. Computation for this fraction is detailed in Section 5.3.3. Recalling the solid boundary condition (Equation 5.3) and integrating it along the solid interface yields:

$$\int_{G_{ij} \cap \partial\Omega_S} \boldsymbol{n} \cdot \nabla p^{n+1} \, dA = \frac{\rho}{\Delta t} \int_{G_{ij} \bigcap \partial\Omega_S} \boldsymbol{n} \cdot (\tilde{\boldsymbol{U}}_b - \boldsymbol{U}_b^{n+1}) \, dl. \tag{5.7}$$

Substituting Equation 5.5–Equation 5.7 all into Equation 5.4 and rewriting the equation, we finally obtain:

$$\begin{aligned}&E_{i-\frac{1}{2},j} \cdot \frac{\Delta t(p^{n+1}_{i-1,j} - p^{n+1}_{i,j})}{\rho \Delta x} + E_{i+\frac{1}{2},j} \cdot \frac{\Delta t(p^{n+1}_{i+1,j} - p^{n+1}_{i,j})}{\rho \Delta x} \\ &\qquad + E_{i,j-\frac{1}{2}} \cdot \frac{\Delta t(p^{n+1}_{i,j-1} - p^{n+1}_{i,j})}{\rho \Delta z} + E_{i,j+\frac{1}{2}} \cdot \frac{\Delta t(p^{n+1}_{i,j+1} - p^{n+1}_{i,j})}{\rho \Delta z} \\ &= E_{i+\frac{1}{2},j} \cdot \tilde{u}_{i+\frac{1}{2},j} - E_{i-\frac{1}{2},j} \cdot \tilde{u}_{i-\frac{1}{2},j} + E_{i,j+\frac{1}{2}} \cdot \tilde{w}_{i,j+\frac{1}{2}} - E_{i,j-\frac{1}{2}} \cdot \tilde{w}_{i,j-\frac{1}{2}} - \int_{G_{ij} \bigcap \partial\Omega_S} \boldsymbol{n} \cdot \boldsymbol{U}_b^{n+1} \, dl.\end{aligned} \tag{5.8}$$

The above equation provides a symmetric positive definite linear system for pressure computation, and in this study it is solved by the BCG method described in Press et al. (1992).

For fixed structures or structures with prescribed motions (e.g. the wavemaker), the boundary velocity $\boldsymbol{U}_b^{n+1}$ in Equation 5.8 is known, and hence Equation 5.8 can be easily solved as all terms at the RHS are knowns. For freely moving structures, the solutions are more complicated as $\boldsymbol{U}_b^{n+1}$ is unknown. In weak fluid-structure coupling schemes, $\boldsymbol{U}_b^{n+1}$ may be replaced by $\boldsymbol{U}_b^{n}$ to solve Equation 5.8. Once the fluid pressure field is resolved, an integration of fluid pressure on the structure boundary is usually carried out to calculate the fluid force on the structure, and then move the structure according to Newton's second law of motion. However, when implementing this weak coupling scheme with the current pressure projection method, it is found that the motion of freely moving structures can become very unstable. This is due to the fact that the pressure is used to both project the velocity field and advect the structure; a small error in predicting the structure velocity will cause a significant pressure change in the fluid, which in turn leads to an incorrect structure motion, and vice versa. Instead of iterating the whole scheme to improve the stability, as was done in Hadžić et al. (2005), here an efficient strong coupling algorithm is proposed. This is achieved by reinterpreting the technique proposed in Batty et al. (2007), where the structure velocity is purely expressed in terms of the pressure, using a J operator that maps the pressure to net forces and torques on the structure:

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t M_s^{-1} J p^{n+1} + \Delta t \boldsymbol{f}, \tag{5.9}$$

where $\boldsymbol{U}^{n+1}$ and $\boldsymbol{U}^n$ are the structure velocities at time-step $n+1$ and n, respectively; M_s is the mass matrix of the structure.

The operator J is formed following Batty et al. (2007). For example, the x-component of the translational force on the structure can be written as:

$$F_x = -\iint_{\partial\Omega_S} p\boldsymbol{n}\, dA = -\iiint_{\Omega_S} \nabla p\, dV\,, \tag{5.10}$$

where dV denotes the volume differential. Rewriting the above equation into its

discretisation form in two spatial dimensions gives:

$$F_x \simeq -\sum_{i,j} V_{i+1/2,j} \frac{p_{i+1,j} - p_{i,j}}{\Delta x}\,, \tag{5.11}$$

where $V_{i+1/2,j}$ is the cell volume of velocity cell that is occupied by the structure; computation of the cell volume is discussed in Section 5.3.4. Via rewriting Equation 5.11 the x-translational part of the operator J is obtained:

$$J_{1,(i,j)} = \frac{V_{i+1/2,j} - V_{i-1/2,j}}{\Delta x}\,. \tag{5.12}$$

The z-translational part is formed in the same manner. Similarly, the torque on a structure can be expressed as:

$$M_t = -\iint_{\partial\Omega_S} (\boldsymbol{r} - \boldsymbol{r}_c) \times p\boldsymbol{n}\, dA = \iiint_{\Omega_S} \nabla p \times (\boldsymbol{r} - \boldsymbol{r}_c)\, dV\,, \tag{5.13}$$

where $\boldsymbol{r}$ is the point of action and $\boldsymbol{r}_c$ is the rotation centre of the structure. Similar to the translational part, the torque part of the operator J is finally obtained as:

$$J_{3,(i,j)} = -\frac{V_{i,j+1/2} - V_{i,j-1/2}}{\Delta z}\,(x_i - X_c) + \frac{V_{i+1/2,j} - V_{i-1/2,j}}{\Delta x}\,(z_j - Z_c)\,, \tag{5.14}$$

where (x_i, z_j) and (X_c, Z_c) are the locations of the action point and the rotation centre, respectively.

Once the operator J is organised, it is straightforward to show that the structure velocity $\boldsymbol{U}^{n+1}$ is actually an explicit expression of the pressure via Equation 5.9–Equation 5.14, which means that the velocity integral at the RHS of Equation 5.8 can also be expressed as a function of the pressure in cells immediately surrounding the structure, as the velocity at any point on the structure surface can be computed by:

$$\boldsymbol{U}_b^{n+1} = \boldsymbol{U}_t^{n+1} + \boldsymbol{U}_w^{n+1} \times \boldsymbol{R}\,, \tag{5.15}$$

where $\boldsymbol{U}_t^{n+1}$ and $\boldsymbol{U}_w^{n+1}$ are the translational and the angular velocities of the structure at time-step $n+1$, respectively; $\boldsymbol{R} = \boldsymbol{r} - \boldsymbol{r}_c$ represents a vector pointing from the rotation centre to the boundary point. As mentioned above, the structure boundary is discretised into a set of boundary points, which also forms a set of small line segments with an equal length Δl. Therefore, in a single cell (i, j) the velocity

boundary integral of Equation 5.8 can be approximated by:

$$\int_{G_{ij}\bigcap\partial\Omega_S} \boldsymbol{n}\cdot\boldsymbol{U}_b^{n+1}dl \simeq \sum_{n_{ij}} \boldsymbol{n}_k\cdot\left(\boldsymbol{U}_b^{temp}+\boldsymbol{Q}_k(M_s,J,\Delta t,p^{n+1})\right)\Delta l\,, \tag{5.16}$$

where $\boldsymbol{n}_k$ is the outward pointing unit normal vector; $\boldsymbol{U}_b^{temp}$ represents the updated boundary velocity due to $\boldsymbol{U}^n$ and $\boldsymbol{f}$ in Equation 5.9; $\boldsymbol{Q}_k$ denotes the boundary velocity transferred from the pressure immediately surrounding the structure; n_{ij} is the total number of boundary segments located inside cell (i,j). Note that $\boldsymbol{n}_k$, $\boldsymbol{U}_b^{temp}$ and $\boldsymbol{Q}_k$ are all defined at the centre of each boundary segment. $\boldsymbol{U}_b^{temp}$ and $\boldsymbol{Q}_k$ are both calculated/organised according to Equation 5.15, with $\boldsymbol{U}_t^{n+1}$ and $\boldsymbol{U}_w^{n+1}$ being the velocity components due to $\boldsymbol{U}^n+\Delta t\boldsymbol{f}$ (for calculation of $\boldsymbol{U}_b^{temp}$) and $\Delta t M_s^{-1}Jp^{n+1}$ (for organisation of $\boldsymbol{Q}_k$), respectively. n_{ij} is computed at each time step by detecting whether the centre of a segment is located inside the cell or not. This may produce some inaccuracy when a large ratio of Δl to Δx is used. Therefore, in this study a small ratio, 0.2, is utilised. The $\boldsymbol{Q}_k$ term on the RHS of Equation 5.16 connects all the pressures together, and it will be added to the LHS of Equation 5.8, modifying the coefficient matrix of the linear system of equations. The linear system is now not necessarily positive definite or symmetric due to the above manipulation, as the $\boldsymbol{Q}_k$ term changes between cells due to the differing cell volumes occupied by the structure and hence the operator J. However, Press et al. (1992) state that the BCG solver can still be expected to work under these conditions. It is noted that, in Batty et al. (2007), the structure velocity in Equation 5.9 is expressed as the kinetic energy of the structure that is directly added to their energy minimization process. In the scheme presented here, the velocity on the structure boundary is further calculated using Equation 5.16 for the cut cell based finite volume discretisation of the PPE. In addition, a different approach to compute the structure volume fraction V is required due to the fact that the cut cell technique is employed. This is described in Section 5.3.4.

After the pressure field is found on the underlying grid, the structure velocity can be directly updated using Equation 5.9. Alternatively, in this study the pressure on the grid is first interpolated onto the central points of individual boundary segments using the bilinear interpolation, and then the fluid force on each segment is calculated and integrated to obtain the overall force on the structure. Finally, the structure velocity and position are updated using a simple explicit Euler time-advancing scheme. In this way, sub-grid scale accuracy on the calculation of fluid force on the structure

is achieved, compared with using Equation 5.9, which integrates the fluid force on the grid scale. It is noted that during the solution of the PPE, the $\boldsymbol{Q}_k$ term in Equation 5.16 updates the structure velocity using the pressures only in the surrounding fluid cells where the pressure values need to be resolved; therefore, due to the grid discretisation, the hydrostatic case for floating bodies may not be guaranteed, as the fluid cell centres, where the pressure values are stored, are usually not aligned with the structure boundary. To compensate for this, before the interpolation for fluid pressure on the boundary segments is performed, the pressure field at the fluid cells is linearly extrapolated, considering the effect of gravity, into cells fully occupied by structures, where no pressure values are resolved.

5.3.3 Fraction of grid face open-to-water (E)

The fraction of grid face open-to-water, E, is calculated based on the SDF ϕ_s computed at the cell vertices. Figure 5-3 shows a typical cell that is partially occupied by a structure. The fraction open-to-water $E_{i-1/2,j}$ at grid face $(i-1/2, j)$ is computed by:

$$E_{i-1/2,j} = \begin{cases} \Delta x \frac{\phi_{s2}}{(\phi_{s2}-\phi_{s1})} & \text{if} \quad \phi_{s1} < 0 \text{ and } \phi_{s2} > 0 \\ \Delta x \frac{\phi_{s1}}{(\phi_{s1}-\phi_{s2})} & \text{if} \quad \phi_{s1} > 0 \text{ and } \phi_{s2} < 0 \\ 0 & \text{if} \quad |\phi_{s1}| \leq 0.01\Delta x \text{ and } |\phi_{s2}| \leq 0.01\Delta x \text{ and } \phi_{s1}\phi_{s2} < 0 \\ 0 & \text{if} \quad \phi_{s1} < 0 \text{ and } \phi_{s2} < 0 \\ \Delta x & \text{if} \quad \phi_{s1} > 0 \text{ and } \phi_{s2} > 0 \end{cases} \tag{5.17}$$

where Δx is the cell length. The other fractions $E_{i+1/2,j}$, $E_{i,j-1/2}$ and $E_{i,j+1/2}$ are computed in a similar manner. It is noted that in two spatial dimensions, for numerical stability, a correction has also been made after all the fractions have been calculated using Equation 5.17; that is, if any three of the four cell edges in a cell have fractions that are all less than a small number 1.0×10^{-8} (i.e. they are all approximately zero), then the fraction of the fourth cell edge is set to zero.

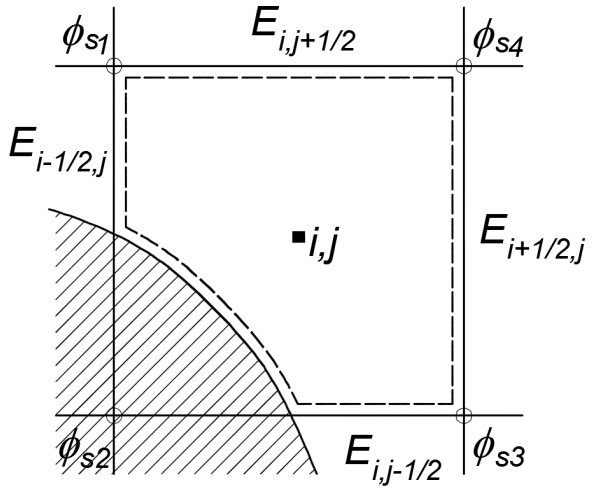


Figure 5-3: Grid face fraction in a staggered grid occupied by solid objects (shadow area).

5.3.4 Volume of cell occupied by structures (V)

In this section both the current method and the method of Batty et al. (2007) used to compute the structure volume fraction V, used in Equation 5.12 and Equation 5.14 for forming the operator matrix J, are presented. Figure 5-4 shows examples of two different types of cut-cell boundaries, where type 1 boundary (left panel) and type 2 boundary (right panel) differ in whether the structure boundary cuts the vertical grid face or not. Within the cell surrounding the horizontal velocity sample $u_{i+1/2,j}$, Batty et al. (2007) use the area of the structure region (gray colour part in Figure 5-4) as the structure volume fraction, which is required by their variational framework for calculating the kinetic energy in cells surrounding velocity samples. Alternatively, in this study, the approach is to use the fraction ratio β of the associated vertical cell edge that is occupied by the structure region and multiply it by the cell area, i.e. $V = \beta(\Delta x)^2$. Here, the structure fraction β is straightforward to compute as it is simply the complement to the fraction open-to-water E mentioned above. Note that for the type 2 boundaries β equals unity in the current approach, as the vertical face $(i + 1/2, j)$ is fully occupied by the solid.

The reason for using this method to calculate the cell volume is that, when the cut cell technique is used for computing fluid pressures, there are no pressure values inside cells fully occupied by structures. In this case, the method of Batty et al. (2007) is not appropriate to be used in the current scheme for converting fluid pressure into

net forces. Using the current approach for calculating the structure volume fraction ensures that the pressures in full structure cells cancel out via Equation 5.12 and Equation 5.14. For example, in a type 2 boundary, assuming cell $(i+1, j)$ and $(i+2, j)$ is fully occupied by a structure, then $V_{i+3/2,j} = V_{i+1/2,j} = (\Delta x)^2$, which, according to Equation 5.12, results in a zero value for the coefficient $J_{1,(i+1,j)}$ of the operator J, and thus eliminates the requirement of having a pressure value in cell $(i+1, j)$ for the calculation of the horizontal fluid force on the structure.

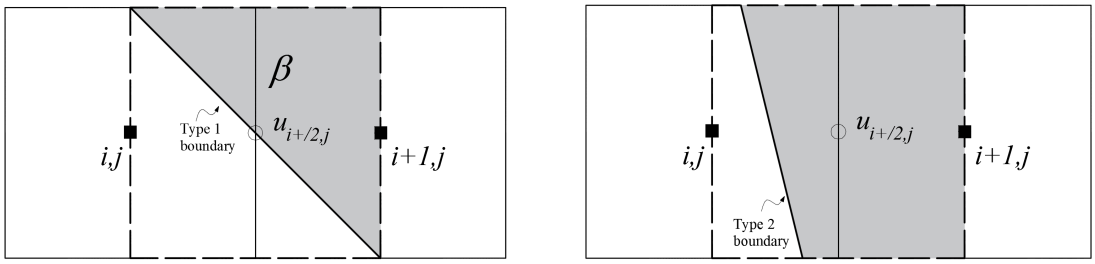


Figure 5-4: Cell volume of structure in velocity cells. The area surrounded by dashed line represents the velocity cell surrounding the velocity sample $u_{i+1/2,j}$, with structure region indicated by the gray colour.

5.4 Numerical algorithm

The solution procedure of the "full particle" PIC model, incorporating the Cartesian cut cell based two-way strong fluid-structure coupling algorithm, is summarised in Figure 5-5. As in chapter 3, the variables adjusted at each step of a computational cycle are also indicated.

5.5 Numerical wave tank (NWT)

5.5.1 Wave generation and absorption

The setup of a NWT is detailed in this section. Waves are generated by a piston-type wave paddle at the upstream end of the tank and absorbed via a relaxation zone

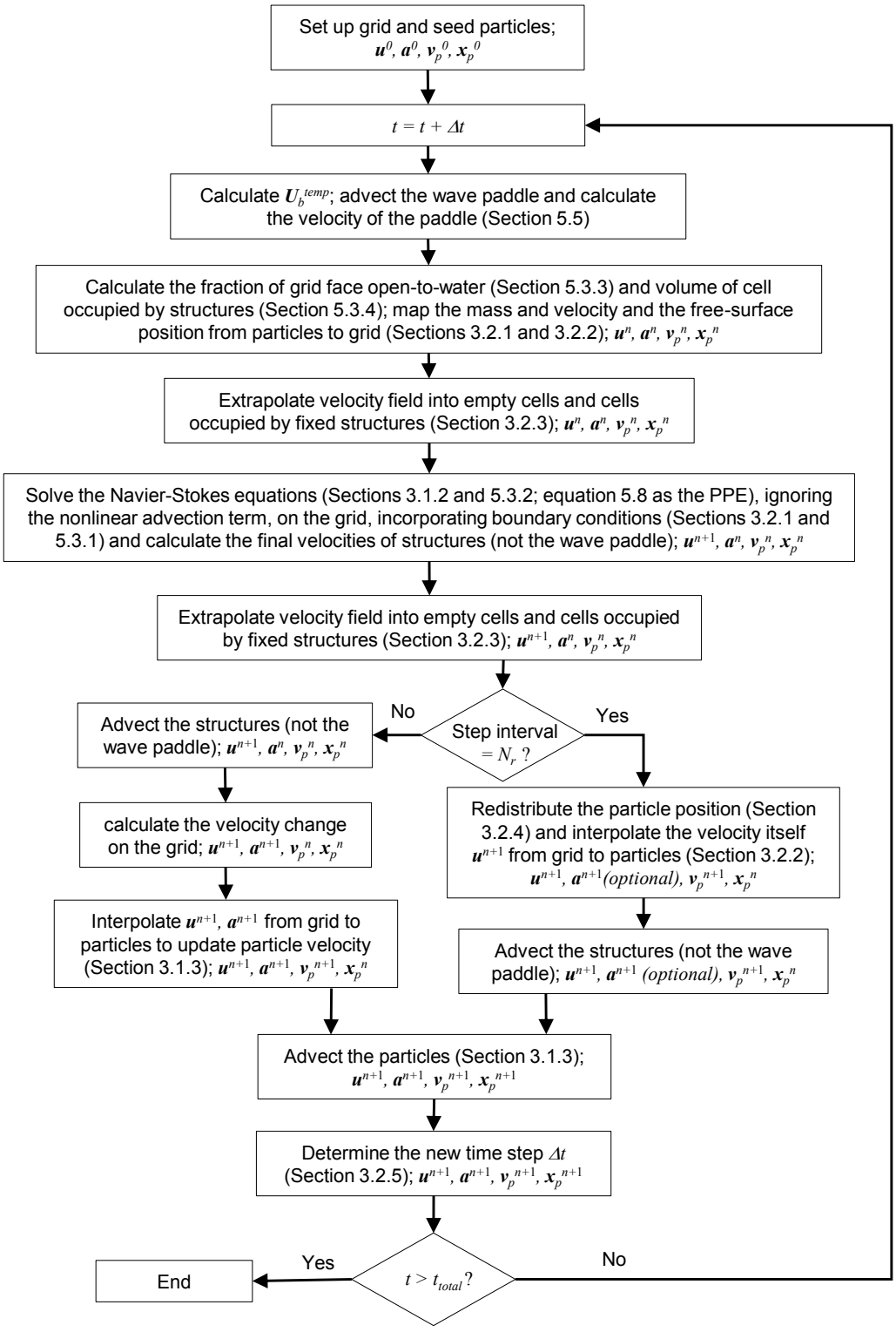


Figure 5-5: Flow chart of the solution procedure of the "full particle" PIC model, incorporating the cut cell based method for fluid-structure interaction.

method at the downstream end of the tank. The wavemaker moves in the x-direction. Linear wave theory (Dean and Dalrymple, 1991) is used for wave generation by giving a prescribed velocity wall boundary that is resolved via Equation 5.8. The relaxation zone approach proposed in Mayer et al. (1998) and Jacobsen et al. (2012) is employed to damp the velocity carried by particles for wave absorption. This section focuses on the discussion of regular wave generation and absorption, and the generation of other types of waves using the wave paddle are introduced in the following sections.

For regular wave generation, the paddle velocity is prescribed as:

$$U_b(t) = \frac{H}{2\Psi}\omega \cos\omega t \tag{5.18}$$

where:

$$\Psi = \frac{4\sinh^2 kh}{2kh + \sinh 2kh}, \tag{5.19}$$

where H, ω and k represent the target wave height, wave frequency and wave number, respectively, and h accounts for the water depth. The displacement of the wave paddle is prescribed simply as $\xi(t) = \int_0^t U_b(t)\,\mathrm{d}t$.

For wave absorption, the velocity carried by particles, entering the absorption zone at the downstream end of the NWT in x-direction, is forced to be the desired analytical solution, which in our case is water at rest:

$$\boldsymbol{v}_r = R_a(x_p)\boldsymbol{v}_{ana} + (1 - R_a(x_p))\boldsymbol{v}_{num}, \tag{5.20}$$

where $\boldsymbol{v}_{ana} = [0, 0]^T$ is the desired analytical velocity and $\boldsymbol{v}_{num}$ is the computed numerical velocity $[v_x, v_z]^T$, where v_x and v_z are the velocity components in x- and z-directions respectively; $R_a(x_p)$ is the relaxation coefficient, which is a function of the particle location in x-direction, x_p,

$$R_a(x_p) = 1 - \frac{\exp(x_r^{3.5}) - 1}{\exp(1) - 1}\,, \tag{5.21}$$

$$x_r = (x_p - x_{sta})/D_L\,, \tag{5.22}$$

where x_{sta} and D_L are the starting location and the length of the absorption zone in x-direction, respectively. The wave absorption is conducted after the particles have been advected. It is noteworthy that this technique is applied in the same manner in three spatial dimensions.

A validation of regular wave generation and absorption has been conducted. A 2D NWT was set up with the piston-type wavemaker placed at $x = 0.625$ m and the starting location of the absorption zone, x_{sta}, being 40 m. The length of the NWT ranges from 45 to 65 m to account for the use of different lengths of the absorption zone. The still water depth h was fixed at 2.5 m. Regular waves with a period of 1.64 s and three different wave heights, $H = 0.1$ m, 0.25 m and 0.4 m, were tested; the wave steepnesses kA ($A = H/2$ is the wave amplitude) are 0.075, 0.188 and 0.300, respectively. The computational grid size was set to $\Delta x = \Delta z = 0.025$ m for all cases, resulting to 100 grid cells being initially set up in z-direction to accommodate the still water depth. The Courant number used to control the variable time step was 0.5. The time histories of wave elevations at three locations, $x = 9.0$, 37.95 and 39.0 m, were recorded during the simulation. While the first one was used to validate the wave generation, the other two were used to calculate the reflection coefficient based on the two-point method of Goda and Suzuki (1976). It may be worth noting that in the numerical simulation the free-surface elevation was calculated by considering the level of the highest particle at the column.

Figure 5-6 plots the numerical predictions of the time histories of wave elevation at $x = 9.0$ m ($3.35h$ from the wavemaker), in comparison with the theoretical results. From Figure 5-6, it can be seen that the numerical results are in good agreement with the theoretical values for wave steepnesses kA =0.075 and 0.188. However, for $kA = 0.3$, where the wave nonlinearity is very strong, a relatively large discrepancy is shown, particularly around the wave troughs.

For the validation of wave absorption, the computed reflection coefficients against the ratio of the absorption zone length D_L and the wave length L are plotted in Figure 5-7. The reflection coefficient is calculated as the ratio of the incident and reflected wave amplitudes of the first harmonic. Twelve different absorption zone lengths were investigated for all the three wave steepness conditions. Note that here the wave steepness is calculated based on the wave amplitude of the incident wave of the first harmonic, A_I, that is separated from the wave elevations recorded at $x =$ 37.95 and 39.0 m by the two-point method of Goda and Suzuki (1976). As shown in Figure 5-7, the wave steepnesses here are slightly smaller than the input ones, indicating that there may be numerical dissipation involved in the numerical modelling of wave propagation. Nevertheless, from Figure 5-7, it is seen that, generally, as the wave steepness increases, the reflection coefficient becomes larger. This trend

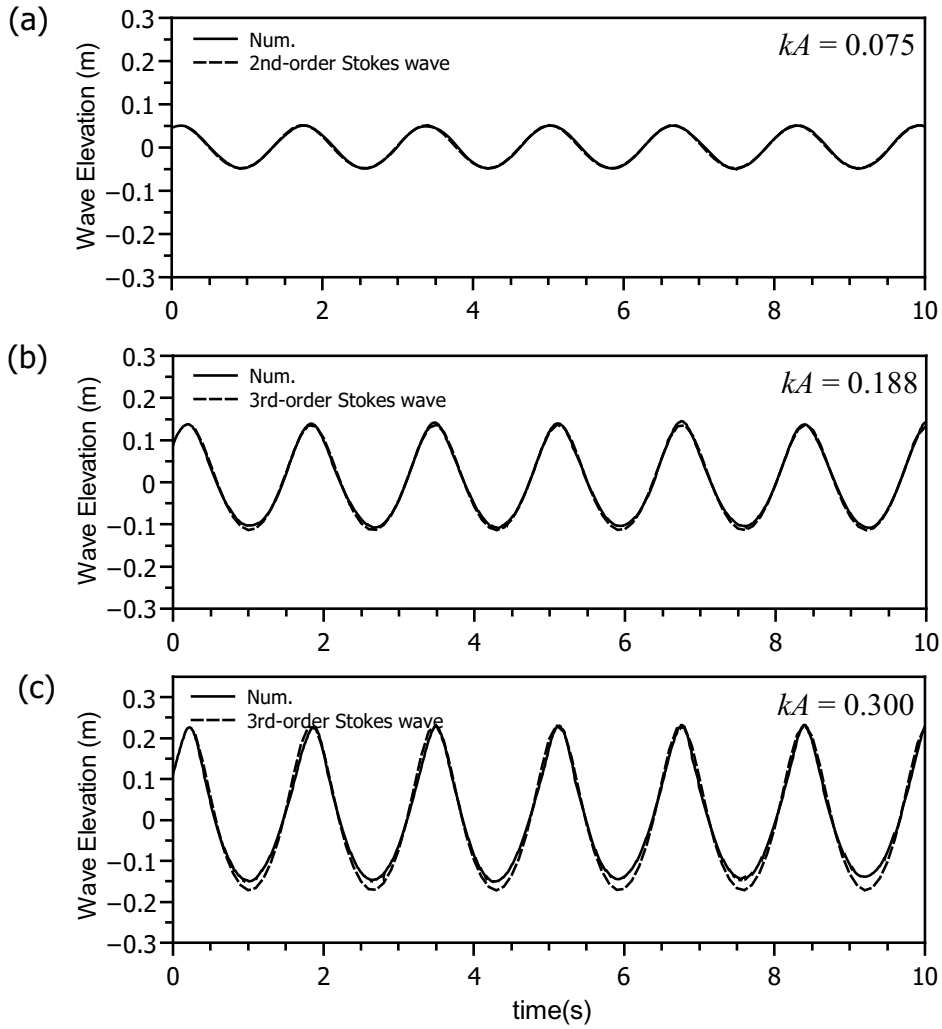


Figure 5-6: Comparison of free-surface elevations between the numerical predictions at $x = 9.0$ m and the analytical solutions.

is consistent with what has been reported by other researchers for the relaxation zone method (Jacobsen et al., 2012; Dimakopoulos et al., 2016) and shows that the current implementation of this approach in the "full particle" PIC solver works best for weakly nonlinear (and linear) waves. It may be also seen from Figure 5-7 that generally the relaxation zone approach used here performs well and is probably most cost-effective when the absorption zone length is around 2–4 wavelengths. Particularly, for the smallest wave steepness presented, the reflection coefficient is around 1%. This is relatively close to what has been reported by Dimakopoulos et al. (2016). It is noteworthy that currently only the particle velocity is relaxed. To improve the performance of wave absorption, the particle position could also be relaxed to a specified value, akin to the volume fraction relaxation suggested by Jacobsen et al. (2012).

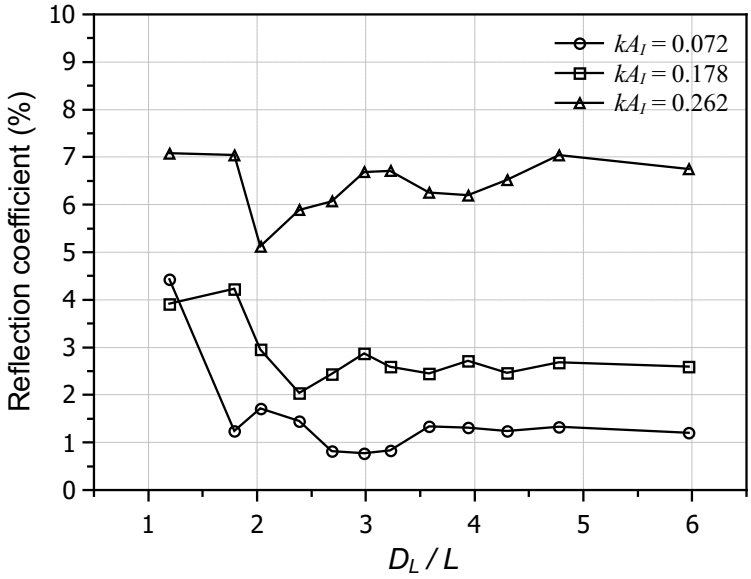


Figure 5-7: Reflection coefficients for different absorption zone lengths and wave steepnesses.

5.5.2 Focused wave generation using a wave paddle

For a focused wave the free-surface elevation can be expressed following Zhao and Hu (2012):

$$\eta(x,t) = \sum_i a_i \cos(k_i(x - x_f) - \omega_i(t - t_f)) \,, \tag{5.23}$$

where,

$$a_i = A_f \frac{G(f_i)}{\sum_i G(f_i)} \,, \tag{5.24}$$

k_i and f_i are the wave number and wave frequency for the ith wave component, respectively; $\omega_i = 2\pi f_i$; x_f, t_f and A_f are the focused position, focused time and focused wave crest value, respectively. Equation 5.24 computes the wave amplitude assigned to each wave component, according to a wave power spectrum S_p. In this equation, the calculation of $G(f_i)$ can significantly affect the shape of the generated focused wave. While Chen et al. (2014b) use the wave energy spectrum directly:

$$G(f_i) = S_p(f_i)\Delta f \ , \tag{5.25}$$

Zhao and Hu (2012) adopt the amplitude value representing the energy:

$$G(f_i) = \sqrt{2S_p(f_i)\Delta f} \ . \tag{5.26}$$

In this study, $G(f_i)$ is selected depending on the particular test case under consideration.

For the wave spectrum, the Joint North Sea Wave Project (JONSWAP) density distribution form proposed in Goda (1999) is used for all test cases:

$$S_p(f_i) = \beta_J H_{1/3}^2 T_p^{-4} f_i^{-5} e^{[-5/4(T_p f_i)^{-4}]} \cdot \gamma^{\exp[-(f_i T_p - 1)^2/2\sigma^2]} \tag{5.27}$$

with

$$\beta_J = \frac{0.06238(1.094 - 0.01915 \ln \gamma)}{0.230 + 0.0336\gamma - 0.185(1.9 + \gamma)^{-1}}\,, \tag{5.28}$$

and $\gamma = 3.3, \sigma = 0.07(\omega < \omega_p)$ and $\sigma = 0.09(\omega \geq \omega_p)$, where T_p is the peak frequency of spectrum.

With the free-surface elevation determined by Equation 5.23, it is straightforward to the calculate the velocity of the wave paddle at location x_0, according to the linear wavemaker theory (Dean and Dalrymple, 1991):

$$U_b(x_0, t) = \sum_i \frac{\omega_i}{T_r(\omega_i)} a_i \cos(k_i(x_0 - x_f) - \omega_i(t - t_f))\,, \tag{5.29}$$

with

$$T_r(\omega_i) = \frac{4\sinh^2 k_i h}{2k_i h + \sinh 2k_i h}\,, \tag{5.30}$$

where T_r is the transfer function. The position of the wave paddle at each time-step is given by an equation formed by integrating Equation 5.29 with respect to time.

5.5.3 Solitary wave generation using a wave paddle

The approach proposed in Wu et al. (2014) is employed to generate solitary waves using the wave paddle. This technique adopts the formula of wave paddle motion proposed by Goring (1978), and augments it by the ninth-order solitary wave solution of Fenton (1972). With the assumption that the average horizontal water particle velocity, adjacent to the wave paddle, equals the wave paddle velocity, the motion of the wave paddle can be expressed as:

$$\frac{d\xi}{dt} = U_b(\xi, t) = \frac{C\eta|_{x=\xi}}{h + \eta|_{x=\xi}}\,, \tag{5.31}$$

in which ξ is the position of the wavemaker at time t; C is the wave celerity; η is the free-surface elevation, and h is the still water depth. According to the ninth-order solitary wave solution of Fenton (1972), the free-surface elevation η at the wavemaker position is given as:

$$\eta = h \sum_{i=1}^{9} \eta_i \left(\frac{H}{h} \right)^i , \tag{5.32}$$

where H is the target wave crest height and η_i is a coefficient associated with $S = \text{sech}[KX]$, where

$$X = \xi - Ct - \xi_0 \tag{5.33}$$

$$K = \sqrt{\frac{3H}{4h^3}} \left(1.0 + \sum_{i=1}^{8} K_i \left(\frac{H}{h} \right)^i \right) \tag{5.34}$$

$$C^2 = gH \left(1.0 + \sum_{i=1}^{9} C_i \left(\frac{H}{h} \right)^i \right) , \tag{5.35}$$

where ξ_0 is the initial position of the wave crest, which must be set to a negative value, as the solitary wave is generated in a quiescent water flume; K_i and C_i are coefficients. The selection of ξ_0 will be based on the wave length (Wu et al., 2014), and the formulas required to determine coefficients η_1 to η_9, K_1 to K_8 and C_1 to C_9 are listed in Table 1 of Fenton (1972) and are not repeated here.

5.6 Test cases

In this section a number of test cases are presented in which the fluid-structure interaction scheme, as well as the "full particle" PIC model, are validated against experimental data and other numerical results. Emphasis is given to the validation of the new scheme for hydrodynamic processes occurring in the vicinity of coastal and offshore structures by five benchmark cases, and these are: i) Overtopping over a low-crested structure (LCS) (Oliveira et al., 2012), ii) Shoaling over a submerged bar (Ohyama et al., 1995), iii) Wave forces on a fixed submerged cylinder (Dixon et al., 1979), iv) 1-DoF roll motion of a rectangular box under regular wave conditions (Jung et al., 2006), and v) 2-DoF floating structures under extreme wave conditions (Zhao and Hu, 2012). All test cases were run on an Intel Core i7-4930K (CPU @3.40Ghz) desktop with 32Gb RAM employing a Windows OS.

5.6.1 Overtopping over a low-crested structure

In Oliveira et al. (2012), a physical model of wave overtopping for regular non-breaking waves over a simple, low-crested, impermeable coastal structure is presented and results are used to validate the Particle Finite Element Method (PFEM). The experiment was performed at the Maritime Engineering Laboratory of UPC-BarcelonaTech, in a wave flume measuring 18 m long, 0.4 m wide, and 0.6 m deep, using a piston-type paddle to generate regular waves. The water depth was set to $h = 0.19$ m. More details on the physical modelling procedure can be found in Oliveira et al. (2012). The initial conditions used for the experiments and the numerical simulations are shown in Figure 5-8, along with the location of the five free-surface elevation probes. The distances between the wave probes (0–5) and the wavemaker are (in metres) 3.00, 6.60, 6.95, 7.42, 8.69 and 8.79. In the physical model, two cases were tested corresponding to a group of four regular waves with period $T = 1.55$ s and wave heights of $H = 0.06$ m and $H = 0.07$ m, respectively. In both cases, the first and the last wave were linearly ramped and the two middle waves were generated at full height. The wave generation set-up is replicated at the present model for the lower wave height case ($H = 0.06$ m), using the following displacement function for the wave paddle:

$$\xi_R(t) = \begin{cases} \xi(t)\frac{t}{T} & 0 < t \leq T \\ \xi(t) & T < t \leq (N-1)T \\ \xi(t)(1 - \frac{t-(N-1)T}{T}) & (N-1)T < t \leq NT \end{cases} \tag{5.36}$$

with N being the desired number of waves, which is set to 4 for this test case. In addition, following Oliveira et al. (2012), an 8.2% reduction of paddle displacement was applied in the numerical model to account for the leakage between the paddle and the walls of the flume in the physical experiments. Figure 5-9 shows the paddle displacement used in the current numerical simulation.

For the numerical modelling, three different sets of grid size were used and these being: $\Delta x = \Delta z = h/19$ (Case A), $h/38$ (Case B) and $h/76$ (Case C), with about 67,000, 270,000 and 1 million particles, respectively. The simulation was run for 20 s (Courant number = 0.5) and required around 21 min of CPU time for Case A, while, with similar grid configuration, the PFEM model of Oliveira et al. (2012) required 50 hours of CPU time on a 2.67GHz Intel Core i7 CPU920. The current model is

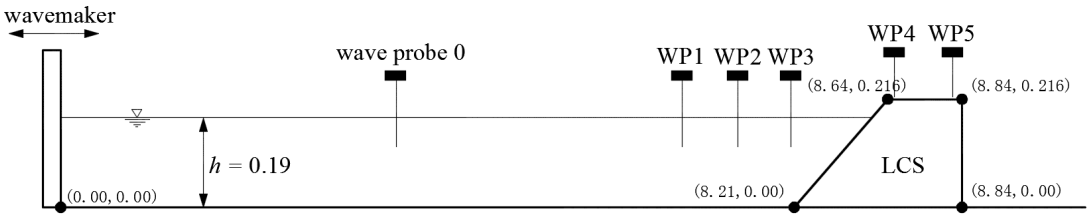


Figure 5-8: Schematic showing the initial conditions and wave probe locations for test 1; all dimensions in (m).

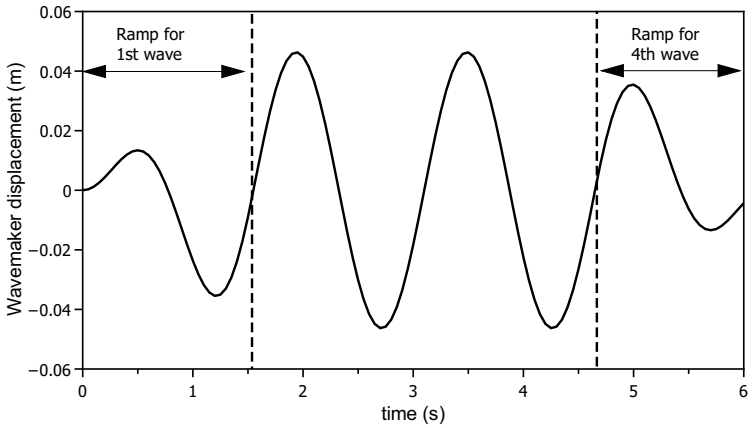
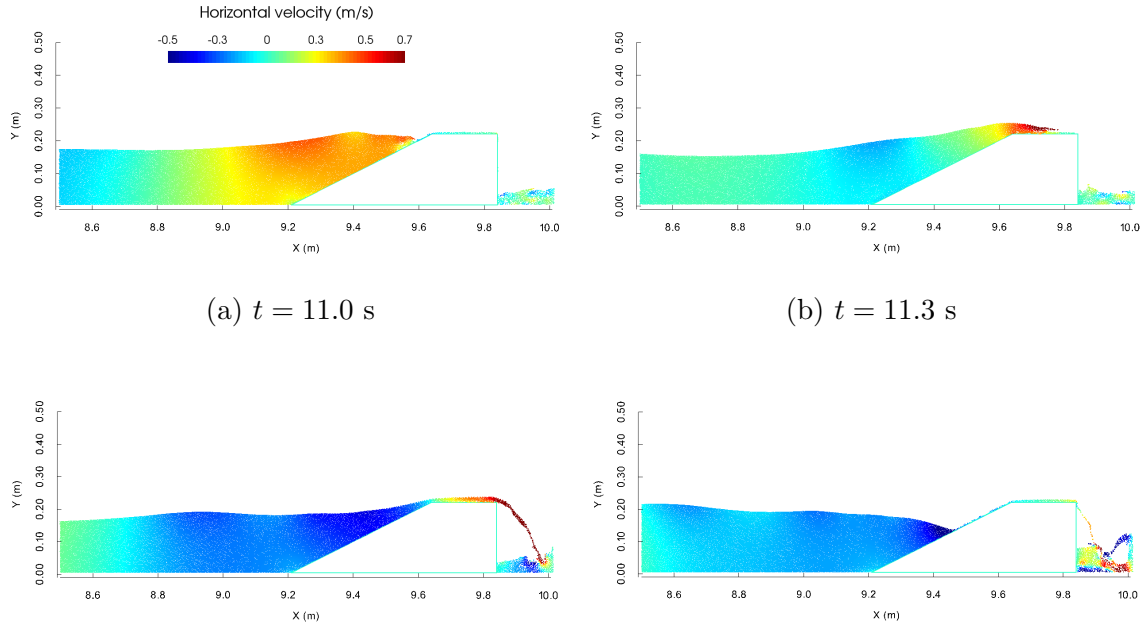


Figure 5-9: Wavemaker displacement for test 1 with $H = 0.6$ m and $T = 1.55$ s.

thus, in this test case, significantly more efficient. This may be explained in two aspects. Firstly, it should be noted that the time step in the current computation for Case A generally varies from 0.002 s to 0.012 s, while the maximum time step used in the PFEM model of Oliveira et al. (2012) was 0.001 s. This means that more numerical steps were conducted in the simulations of Oliveira et al. (2012). Secondly, it is noted in Oñate et al. (2013) that the PFEM requires iterations (typically 3 iterations per time step) for solving the governing equations and special techniques for regenerating the finite element mesh (as the nodes of the mesh are advected in a Lagrangian fashion as the simulation progresses). These could result in a longer computational time per time step for the PFEM model, as none of these is a necessity in the current PIC model.

In the current numerical simulations the water density is set to 1000 kgm^{-3} and the dynamic viscosity is set to $\mu = 1.0 \times 10^{-3}$ kgm^{-1}s^{-1}; these values were used for all the test cases in this chapter.

Four snapshots of the flow field of Case B are shown in Figure 5-10 during run-up and



(a) $t = 11.0$ s (b) $t = 11.3$ s

(c) $t = 11.6$ s (d) $t = 11.9$ s

Figure 5-10: Wave profile and horizontal velocity field near the LCS during overtopping for the second main wave train.

run-down of the second wave at the LCS, where the horizontal velocity field is also illustrated. It is noted that the current "full particle" PIC model predicts wave run-up and overtopping naturally using the particles in a similar fashion to the meshless SPH method. The velocity field provides some further insights on the overtopping jet evolution such as the jet reversal during run-down, which shows that, in this test case, not all the water volume rising above the water level eventually overtops the structure.

Figure 5-11 presents a grid refinement study, where the time histories of water surface elevations at wave probes 0, 2 and 4 are compared for Case A, Case B and Case C. It is seen that the results are convergent, although the results at WP0 and WP2 of Case A show a slight reduction of the wave crest height and a phase lag, compared with those of Cases B and C. This is likely due to numerical dissipation induced by the relatively coarse grid. It is noted that taking the result of Case C (the finest grid) as the reference solution, the root mean square errors (RMSE) of the results of Cases A and B, normalised by the wave height, are 3.2% and 0.9% (for WP0), 4.5% and 1.7% (for WP2), and 4.0% and 2.6% (for WP4), respectively.

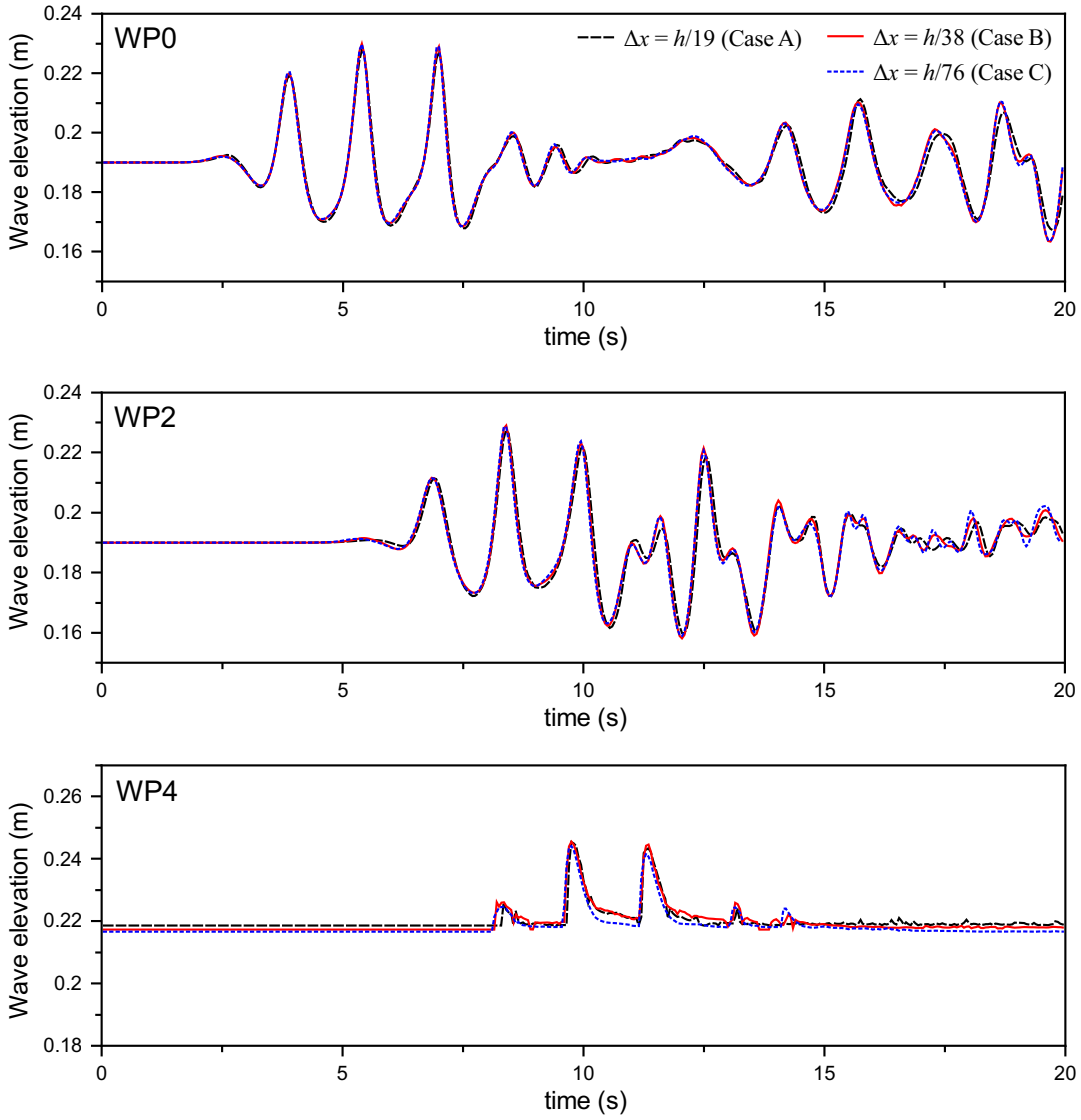


Figure 5-11: Comparison of free-surface elevation at WP0 (top), WP2 (middle) and WP4 (bottom) for three different mesh refinements.

Quantitative comparisons for wave transformation and wave overtopping are shown in Figure 5-12, where the present predictions from Case B are compared with the experimental data as well as the numerical results from the PFEM model of Oliveira et al. (2012). From Figure 5-12, it can be seen that generally, the current results are in good agreement with the experimental observations and the numerical predictions from the PFEM model, although the present numerical model tends to over-predict the wave crests, which may be due to that the incident wave train is slightly over-predicted as seen from the comparison for WP0. From the measurements of the first four wave probes (WP0–WP3), it is shown that the current numerical model

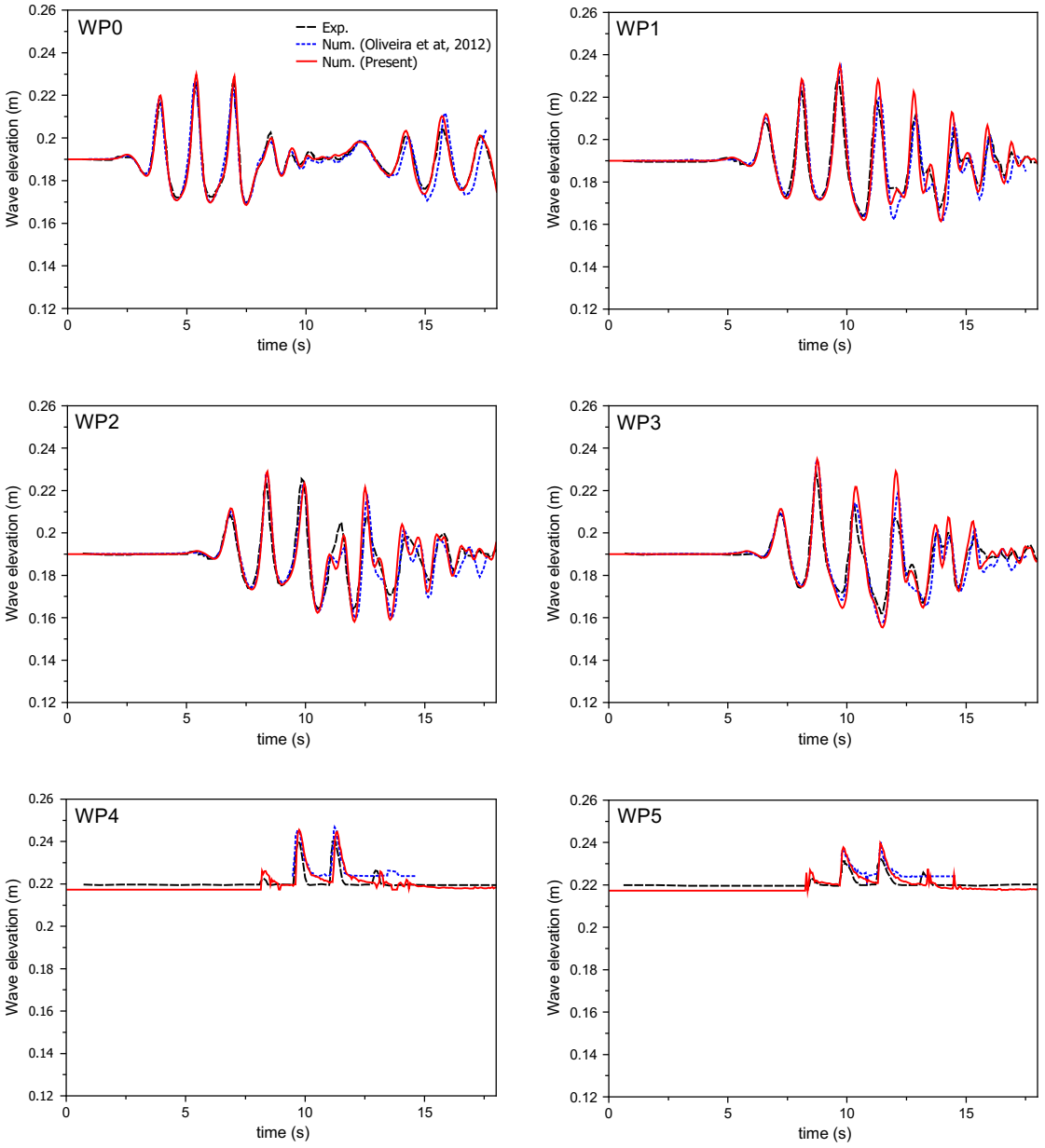


Figure 5-12: Comparison of free-surface elevation at all the six wave probes for the present predictions, experimental data and numerical results from the PFEM model of Oliveira et al. (2012).

reproduces well the nonlinear effects of wave generation, wave propagation, and wave transformation induced by the LCS.

The free-surface elevations recorded at WP4 and WP5 (Figure 5-12) correspond to the thickness of the overtopping jet. In general, the present numerical model

predicts the overtopping jet evolution very well. It is noted that both numerical models slightly overpredict the peak values of the overtopping jet thickness. For the current model, this is probably due to the fact that, processes, such as air resistance and turbulence, which could cause a mild reduction of wave run-up over smooth slopes, are not included in the numerical model. A slight phase delay can also be observed, especially for the first and last overtopping peaks, which is probably due to the lack of resolution in the simulations where the thin layer of water on the top of LCS is not well resolved. During transition from the front to the back of the LCS crest, the numerical model predicts a decrease of the overtopping jet thickness, which is a well-known behaviour (Pullen et al., 2007).

5.6.2 Shoaling over a submerged bar

This test case was used by Ohyama et al. (1995) to test three mathematical models for nonlinear dispersive waves. The authors presented experimental data for various waves propagating over a submerged trapezoidal bar. The setup of the experiment is illustrated in Figure 5-13, where the characteristic dimensions and the location of the wave probes are shown. In the experiment, six wave conditions were tested, emerging from the combination of three wave periods with two wave heights. For the present model, only the larger wave height is considered ($H/h = 0.1$), resulting in a total of three wave conditions, namely Cases 2 , 4 and 6 in Ohyama et al. (1995), with wave periods $T\sqrt{g/h} = 5.94$, 8.91 and 11.88, respectively. In the experiment a piston-type wavemaker was used to generate the waves and a wave absorber is used to remove wave reflection at the end of the flume. The physical wave tank is 65 m long, 1.0 m wide and 1.6 m high and the water depth h was 0.5 m. The distance from the centre of the submerged bar to the wavemaker was 28.3 m.

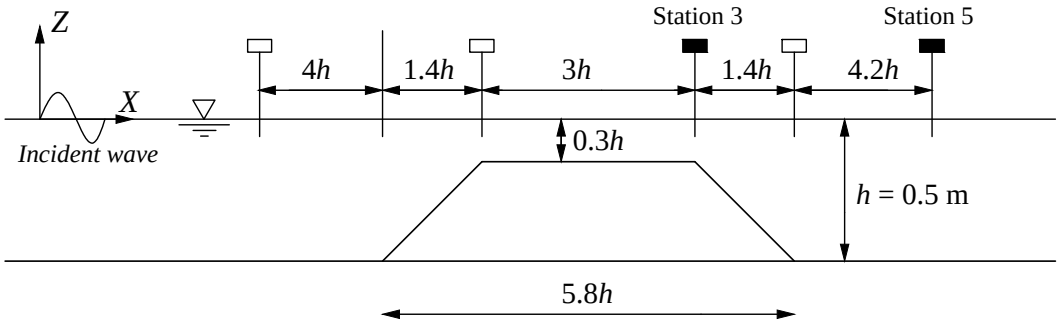
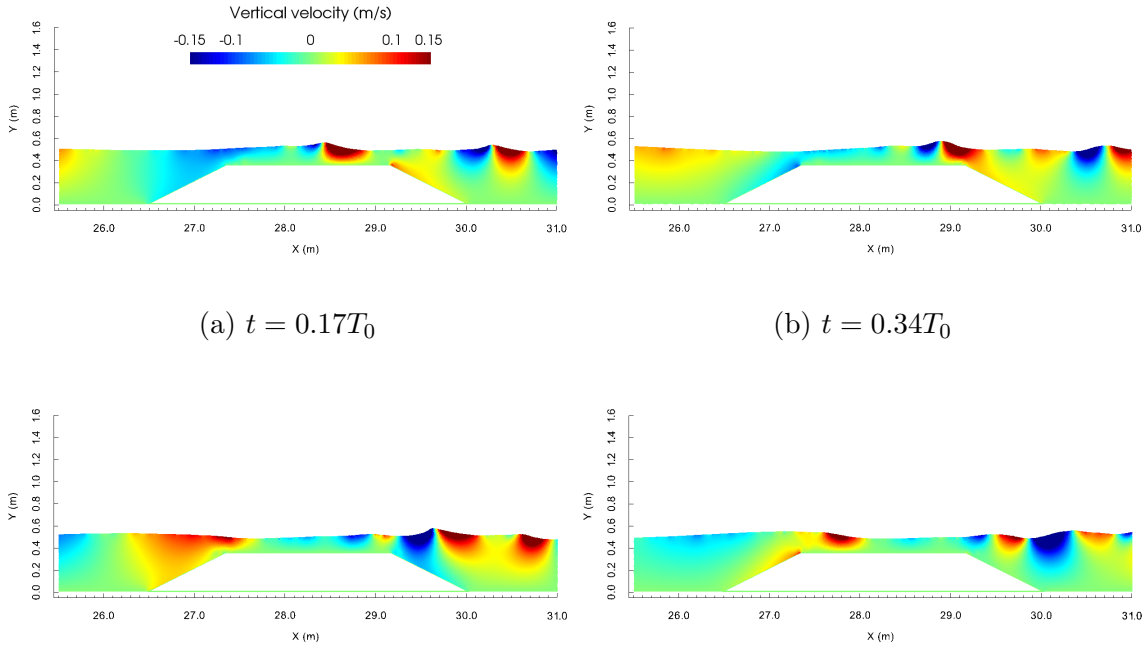


Figure 5-13: Schematic showing the experimental setup for the submerged bar test.



(a) $t = 0.17T_0$

(b) $t = 0.34T_0$

(c) $t = 0.62T_0$

(d) $t = 0.84T_0$

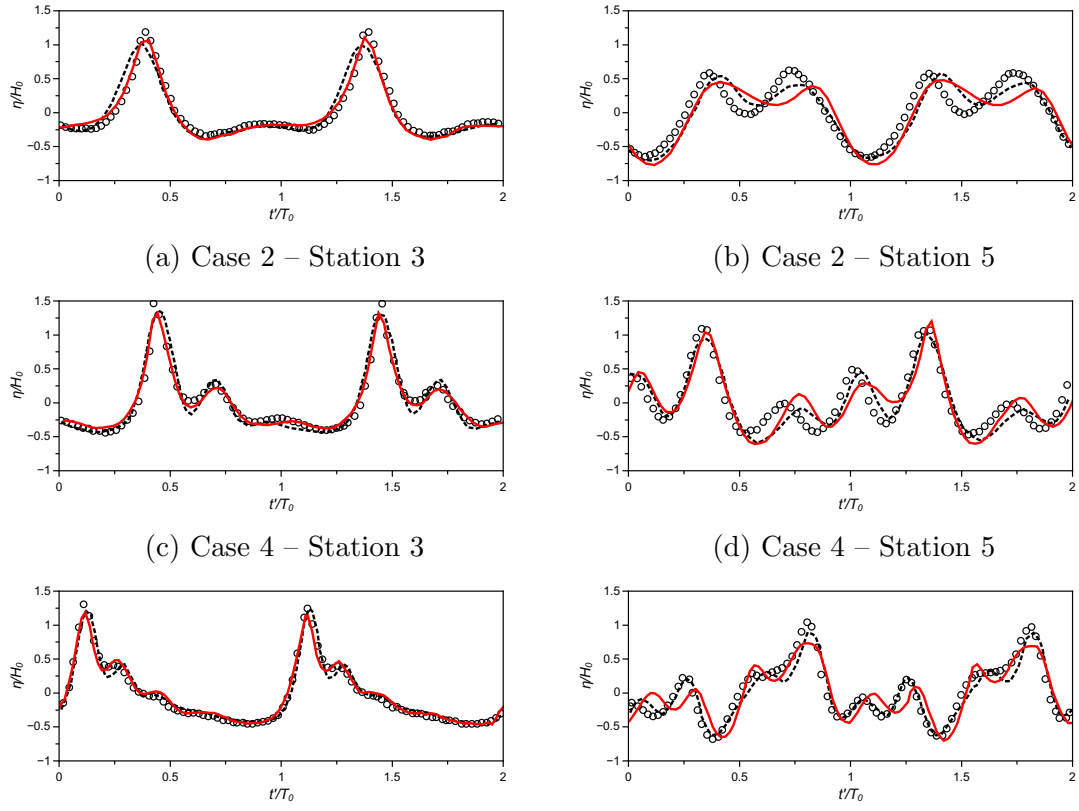
Figure 5-14: Close-up snapshot of wave profile with vertical velocity field in the immediate vicinity of the trapezoidal submerged bar for Case 4.

In the current simulation, while the distance from the centre of the submerged bar to the wavemaker was 28.45 m, which is nearly the same as that used in the experiment, the overall length of the NWT was set to 48 m, 53 m and 58 m for Cases 2, 4 and 6 respectively, in order to reduce the CPU cost. The wave absorption zone at the end of the NWT was at least three wave lengths long for all cases. The grid size was set to 0.01 m $\times$ 0.01 m, resulting to, for example, a cell resolution of 5300 $\times$ 76 with a total number of around 1 million particles for Case 4. The Courant number was set to 0.5 and the run-time was, for instance, approximately 15.1 h for 50 s of simulation time for Case 4.

Figure 5-14 presents snapshots of the numerical simulation, showing the evolution of the waves and the vertical velocity distribution for Case 4, where the nonlinear interaction of the waves with the submerged bar is demonstrated.

Comparison of numerical predictions from the present model and the fully nonlinear model (digitised from Ohyama et al. (1995)) and experimental data of free-surface evolution in time at Stations 3 and 5 (see Figure 5-13) are given in Figure 5-15.

Note that a time shift has been used in order to match the phase at Station 3 for all cases. It is observed that when compared with the experimental data, the present results generally show better agreement at Station 3 than at Station 5; this is further confirmed by the root mean square errors (RMSE), which are around 6% and 21% for all cases at Stations 3 and 5, respectively. The relatively high RMSE at station 5 is primarily due to the phase error, which is likely to be caused by the lack of resolution for the decomposed high-order harmonics when they propagate into the shoreward region. In general, the present model gives results that are similar to those of the fully nonlinear model of Ohyama et al. (1995), and for all the three cases, the present model correctly captures the shape of the wave forms. This is a solid indication that the numerical methodology and the innovative treatment of the nonlinear advection terms through Lagrangian particles is capable of modelling fully nonlinear wave dynamics.



(a) Case 2 – Station 3

(b) Case 2 – Station 5

(c) Case 4 – Station 3

(d) Case 4 – Station 5

(e) Case 6 – Station 3

(f) Case 6 – Station 5

Figure 5-15: Comparison of free-surface elevation at Stations 3 and 5 for Cases 2, 4 and 6. Circles: experimental data; dashed line: results of fully nonlinear model of Ohyama et al. (1995); red line: present results.

Table 5.1: Characteristics of the sub–merged cylinder and incident wave for two different test conditions.

	Test condition 1	Test condition 2
d/D	-0.3	0
A/D	0.2	0.5
kA	0.08	0.2
kh	1.61	1.61

*k is the wave number.

5.6.3 Wave force on a fixed horizontal cylinder

This test case was experimentally studied by Dixon et al. (1979) in order to improve Morison's equation for the calculation of wave forces on fixed, horizontal, partially submerged cylinders. This case was also numerically investigated by Westphalen et al. (2014) using a state-of-the-art ISPH method. The results predicted by the present model were compared with both the experimental and the ISPH data, to assess the performance of the numerical scheme in calculating dynamic pressures and forces.

A cylinder, with diameter D, was initially submerged in still water of depth h. The cylinder submergence, d, is defined as the distance of the centre of the cylinder from the free surface. Regular waves of amplitude A and period T were generated to interact with the cylinder. Two test conditions found in Dixon et al. (1979) and Westphalen et al. (2014) were selected. The key parameters of the tests are shown in Table 5.1, where the cylinder diameter is $D = 1$ m in current simulations.

The non–dimensional vertical wave force $F_z^{'}$ on the cylinder was calculated for comparison:

$$F_z^{'} = \frac{F_z}{1/4\pi D^2 l \rho g}\,, \tag{5.37}$$

where F_z is the measured vertical wave force and l is the length of the cylinder. Note that the vertical wave force F_z is calculated by subtracting the buoyancy force of the submerged cylinder with water at rest from the overall vertical wave force, which, in the experiment, was measured over one cycle starting at the point when the water line would have crossed the still water level going upwards, at the position of the front edge of the cylinder, in the absence of the model.

In the numerical model, regular waves were generated using the techniques presented in Section 5.5 and the length of absorption zone was at least 2.5 times the wave length in both tests (see Table 5.1) to maintain the wave reflection at a low level (see Figure 5-7). The NWT was 150 m long and 5 m high in total, with the cylinder placed at 70 m from the wavemaker. Three sets of computational grid sizes, $\Delta x = \Delta z = D/16$, $D/20$ and $D/25$, were used for a convergence test for the test condition 1, and $D/20$ was chosen to produce data for comparison for both test conditions. This grid configuration resulted in about 958,000 particles. The maximum Courant number was 0.5. It is noted that for the present model it took about 10 h for 66 s of simulation (i.e. 20 wave periods) for test condition 1, while it took the ISPH model of Westphalen et al. (2014) 15 h for 12 s of simulation with 7800 particles.

The convergence test and comparison of the vertical wave force with experimental data and the ISPH model are shown in Figure 5-16, for test condition 1 ($d/D = -0.3$). The current numerical data were sampled for comparison when the waves are fully developed and the vertical wave force reaches the steady state as shown in Figure 5-16(a). Figure 5-16(b) shows the comparison of the results for the three sets of grid sizes. It can be seen that the results are mostly not sensitive to the grid size when $\Delta x = D/20$. It is noted that due to the double grid system of the present model, the use of grid size $D/25$ almost reaches the limitation of the computational resource. Figure 5-16(c) shows the comparison between the numerical results and experimental data. It can be seen that good agreement with the experimental data has been achieved by both numerical models. In particular, compared with the ISPH model of Westphalen et al. (2014), the present "full particle" PIC model predicts better results for both the amplitude and the phase of the vertical wave force. Figure 5-17 shows snapshots of the present numerical results for the free surface and horizontal velocity field during a typical wave cycle for test condition 1 ($d/D = -0.3$).

Similarly, comparison of the vertical wave force and snapshots of the free surface and pressure field for test condition 2 ($d/D = 0$) are shown in Figure 5-18 and Figure 5-19, respectively. For this test condition, the magnitude of the wave force is relatively large as the wave condition is much more violent and the cylinder is above the water surface during most of the wave period. From Figure 5-18, it is seen that the present numerical results agree well with the experimental data and are similar to the ISPH results, demonstrating that, by using the particles, the present

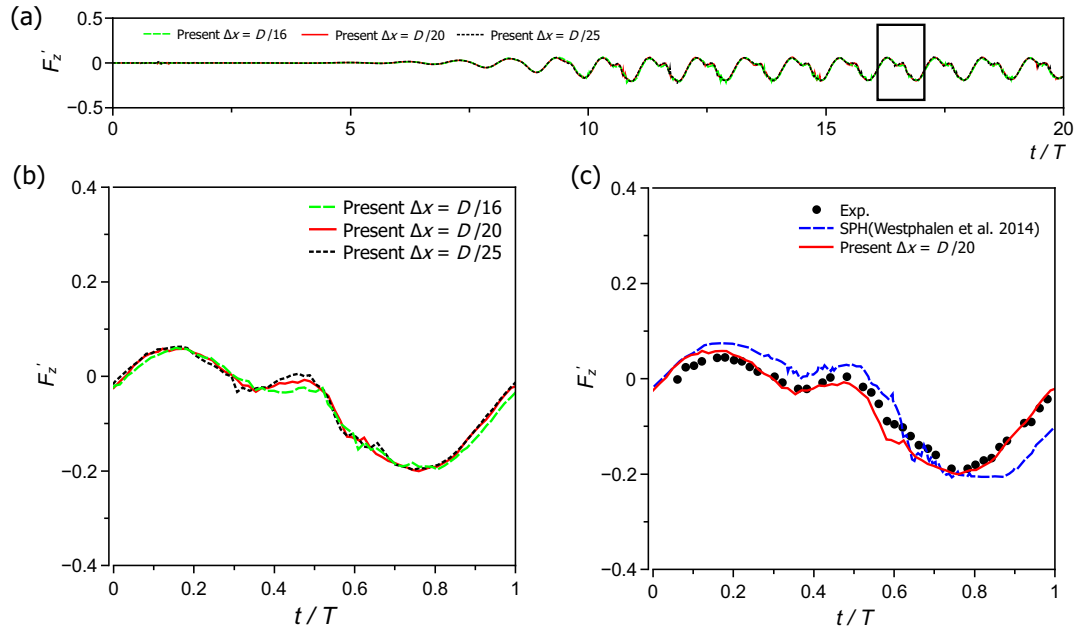


Figure 5-16: Time history of non–dimensional vertical wave forces on the cylinder for test condition 1 ($d/D = -0.3$): (a) numerical data of 20 wave periods from the present model; (b) comparison for results from different grid sizes; (c) comparison between experimental and selected numerical results.

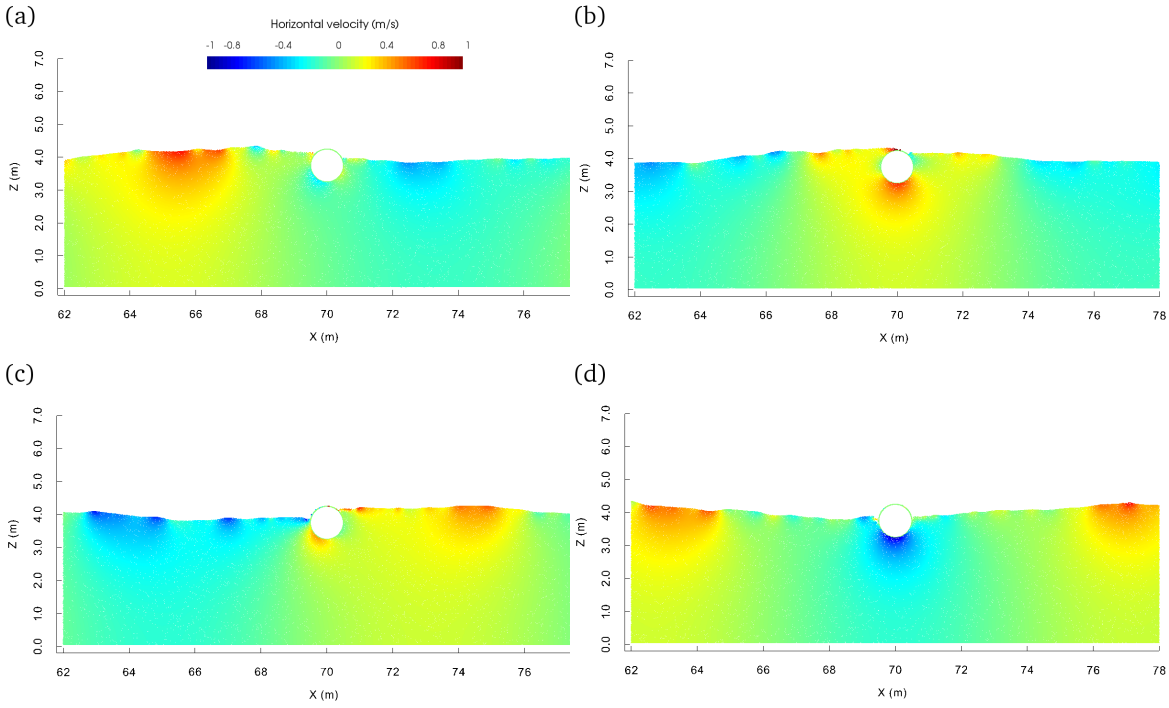


Figure 5-17: Snapshots of the present numerical results for test condition 1 ($d/D = -0.3$), starting from $t = 0.0T$ (panel (a)) with an approximately $0.25T$ time interval.

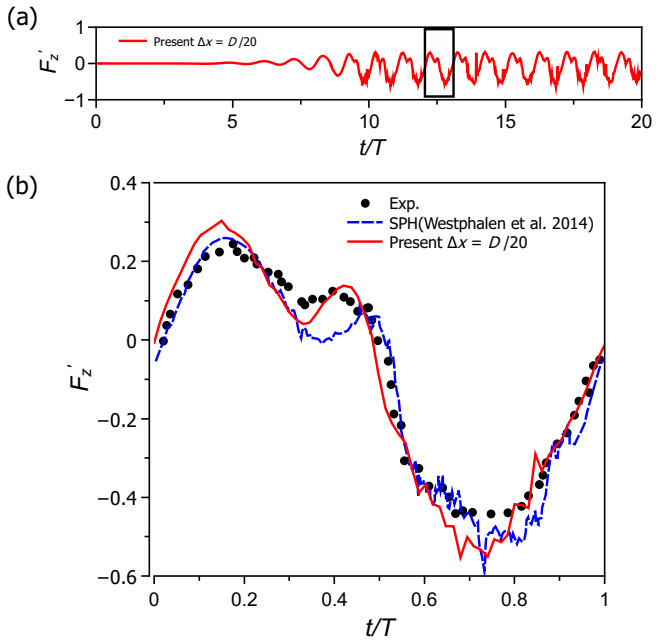


Figure 5-18: Time history of non-dimensional vertical wave forces on the cylinder for test condition 2 ($d/D = 0$): (a) numerical data of 20 wave periods from the present model; (b) comparison between experimental and selected numerical results.

model achieves a similar performance to the meshless ISPH model in handling such a violent wave-structure interaction scenario. In addition, it is worth noting that, due to the use of grid in the current model, very smooth pressure fields have been obtained as shown in Figure 5-19.

In Figure 5-18, the development of spikes in the wave force signal is noted for both numerical models. While this is the case for the meshless ISPH model used by Westphalen et al. (2014), Skillen et al. (2013) demonstrate that this noise can be reduced significantly in ISPH. For the present model, this is probably due to the fact that because a cut cell technique is employed, inaccuracy and instability may occur in cells that are largely occupied by structures, i.e. small cut-cells, particularly during violent fluid-solid interactions. Such small cut-cells are also noted in Kirkpatrick et al. (2003); Qian et al. (2006) and many other works. For further improvements, a cell-merging technique such as that proposed in Kirkpatrick et al. (2003) could be adopted to eliminate the small cut-cells.

In summary, it is demonstrated that the current model is able to reasonably predict wave forces on structures even under violent wave action. Also, compared with the

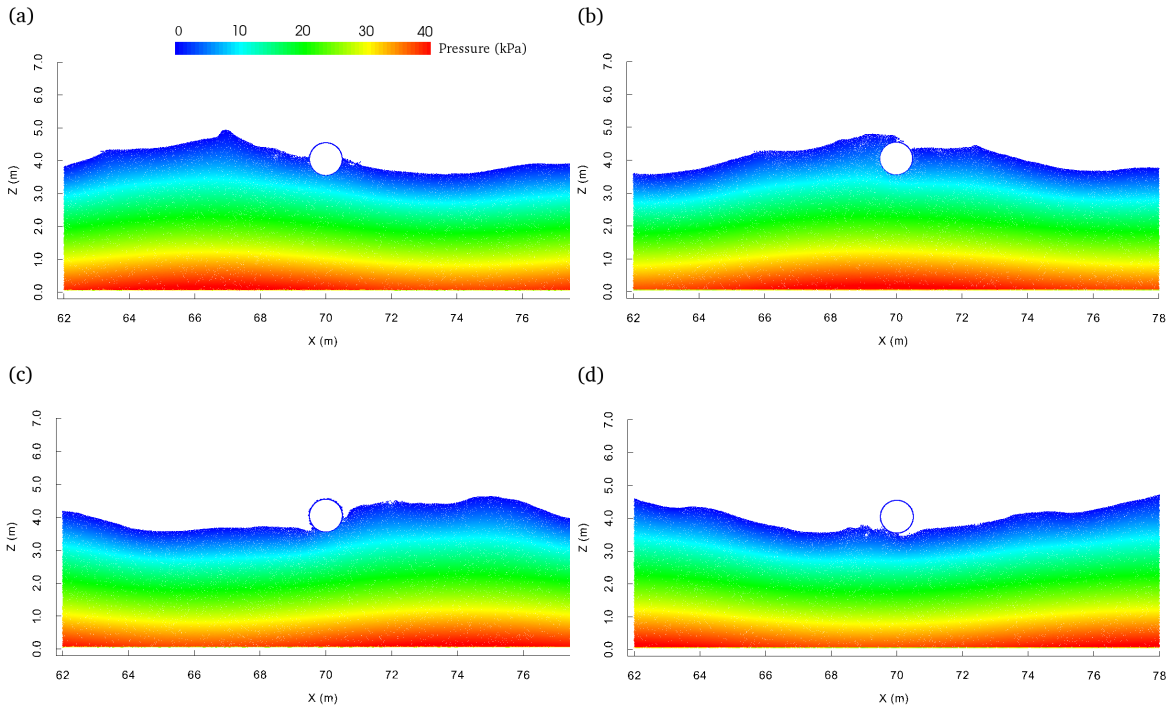


Figure 5-19: Snapshots of the present numerical results for test condition 2 ($d/D = 0$), starting from $t = 0.0T$ (panel (a)) with an approximately $0.25T$ time interval.

ISPH model of Westphalen et al. (2014), the present "full particle" PIC model seems to perform better in predicting the vertical wave force, and is more efficient in terms of the CPU time, which is attributed to the employment of an underlying grid for computational convenience.

5.6.4 Roll motion of a rectangular box

This test case concerns the capability of the current "full particle" PIC model in simulating regular wave interaction with a floating box with roll motion only. This is an example of the situation where viscous damping effects are particularly important in controlling the box motion near the resonance frequency (Downie et al., 1988). To ensure a 1-DoF motion for the floating box, the surge and heave motion of the box and their relevant translation velocities in Equation 5.15 are simply set to zero.

This case was experimentally studied by Jung et al. (2006) with an emphasis on the vortex interaction with the roll motion of the box. The experiments were carried

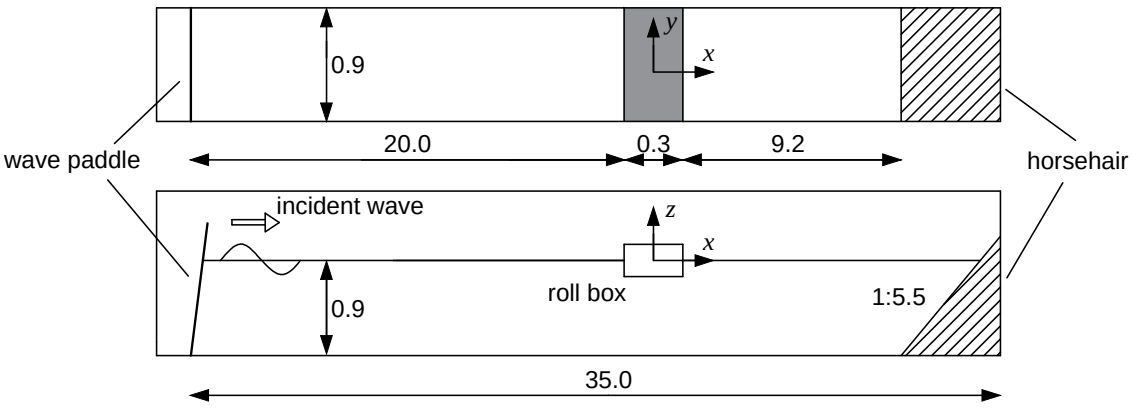


Figure 5-20: Sketch of experimental wave tank in Jung et al. (2006); all dimensions in (m).

out at Texas A&M University and a sketch of the experimental setup is presented in Figure 5-20. The physical model was performed in a 35 m long and 0.9 m wide wave tank, with water depth, h, being 0.9 m. A rectangular box with dimensions: 0.3 m long, 0.1 m high and 0.9 m wide was fixed to the tank and only allowed to rotate along its centre of gravity (0.05 m from the keel), such that the flow can be considered 2D. The draft of the box was 0.05 m and the moment of inertia of the actual mass of the box was $I = 0.236$ kg·m^2. The fluid velocity in the experiment was measured using the particle image velocimetry (PIV) technique. The range of the tested wave conditions for the numerical study were selected according to Jung et al. (2006) and these are shown in Table 5.2.

The NWT was simplified compared to the experimental set-up. First, the horizontal position of the box was not fixed at $x = 20$ m as shown in Figure 5-20, instead, the position of the box, as well as the length of NWT, was changed according to the wave conditions. In particular, for test conditions $1 \sim 5$, the length of the NWT was

Table 5.2: Experimental wave conditions.

Test conditions	Frequency ω(rad/s)	Height H(m)	Wave steepness kA
1	8.98	0.029	0.1191
2	7.85	0.029	0.0912
3	7.39	0.033	0.0919
4	6.76	0.040	0.0931
5	6.28	0.044	0.0887
6	5.24	0.060	0.0849
7	4.49	0.061	0.0653

20 m with the box placed at a distance of 14.5 m from the wavemaker, and for test conditions 6 and 7, the length of the NWT and the distance between the wavemaker and the roll box were 30 m and 20.5 m, respectively. While the box was placed closer to the wavemaker, it was ensured that the roll motion was fully developed before being contaminated by secondary reflections from the wave paddle, as active absorption has not been yet implemented in the present model. The purpose of having a shorter computational domain was to minimise any numerical diffusion that may be present and reduce the CPU cost. An absorption zone as discussed in Section 5.5 was used to absorb waves at the end of the NWT, and this was at least 2.5 times the wave length of each wave condition. The grid size was chosen to be $\Delta x = \Delta z = 0.01$ m (i.e. $h/90$), following the grid refinement study given below, which resulted in total cell numbers ranging from 270,000 to 405,000 for the different test conditions. Because of this, the CPU cost also varied from case to case; for example, it took about 10.8 h for 43 s of simulation (36 wave periods) for the test condition 6 ($\omega = 5.24$ rad/s), which had the maximum 405,000 cells, with 1,061,000 fluid particles.

A free decay test of the box was first carried out in calm water. The rectangular box was initially given an inclination with an angle of 15 degree and then released to roll freely. In the numerical simulation, a flume of 10 m long was used, with the model placed in the middle. This ensured enough time for the development of the decay motion, without being affected by reflected waves. Three different grid sizes were used for a convergence test; these were $\Delta x = \Delta z = 0.02$ m ($h/45$), 0.01 m ($h/90$) and 0.005 m ($h/180$), respectively. The results are given in Figure 5-21(a). It can be seen that the result is nearly convergent when $\Delta x = \Delta z = 0.01$ m ($h/90$).

The comparison between the numerical result and the experimental data for the free decay test is plotted in Figure 5-21(b), from which it is seen that the roll motion is damped out much more slowly in the numerical simulation than in the experiment. It is noted that this smaller damping effect for this case has also been shown in the results of other numerical models; e.g. Nematbakhsh et al. (2013); Ghasemi et al. (2014); Calderer et al. (2014). As argued by Calderer et al. (2014), this is primarily due to the fact that the friction of the experimental apparatus is not considered in the numerical simulation. Calderer et al. (2014) also verified this argument by adding artificial damping in the equation of motion. Also, the present model does not include the strong turbulent dynamics developed during the box roll motion

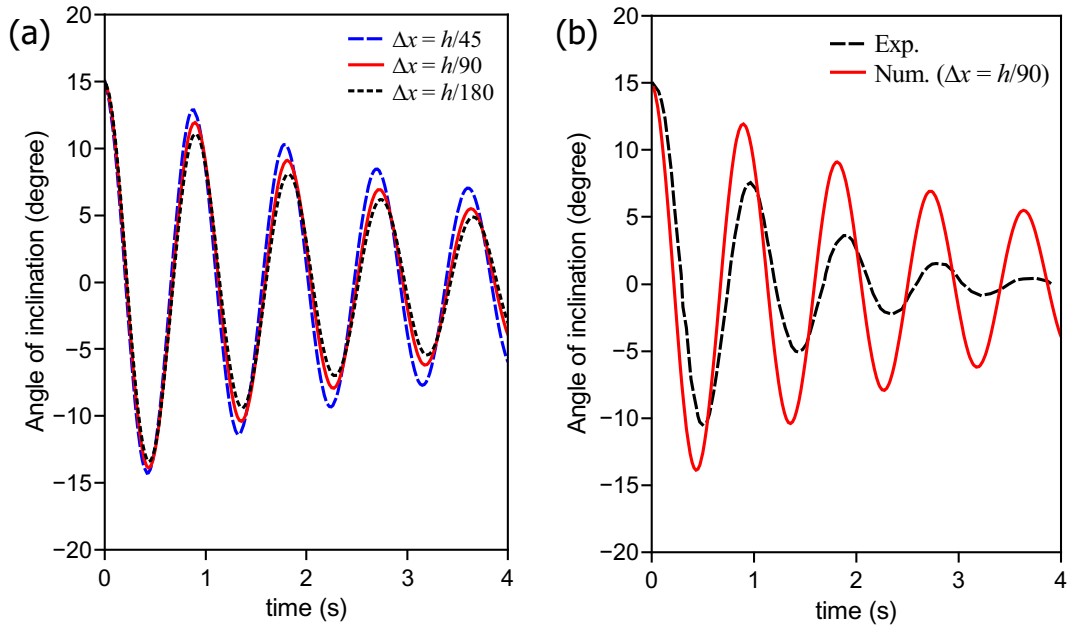


Figure 5-21: Time history of roll motion in the free decay test.

as demonstrated in the experiment of Jung (2005), which will cause more energy dissipation in the system. As a 2D numerical approach, the current model cannot directly resolve the turbulent effects that correspond to the physical processes. In order to resolve turbulent effects for the 2D model, the RANS equations, coupled with a transport model for the turbulence properties (e.g. k-epsilon or k-omega) (Wilcox et al., 1998), could be used for future investigation. Finally, it is noteworthy that from Figure 5-21(a) it can be observed that a more refined grid results in a greater damping effect. This is probably because the boundary layer around the box is better resolved by the fine grid.

Despite the large discrepancy shown in the comparison of the free decay amplitude, the model actually captures the nature frequency of the roll motion with an acceptable accuracy. By averaging the time interval between each peak of the free decay curves given in Figure 5-21(b), the natural frequency from the numerical prediction is $\omega_{num} = 6.90$ rad/s, which is just slightly higher than $\omega_{exp} = 6.78$ rad/s from the physical experiment, with the error being 1.8%.

Figure 5-22 presents the time histories of the predicted roll motion for three typical test conditions. It can be observed that the roll motion of the box reaches steady state after the transition period of first few waves. Figure 5-23 further compares the steady state roll amplitude, magnified by kA under different wave conditions,

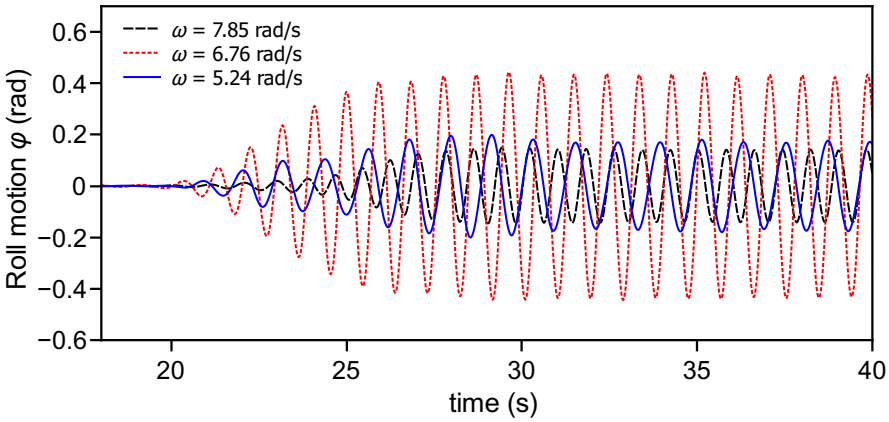


Figure 5-22: Time histories of the roll motion for three wave conditions.

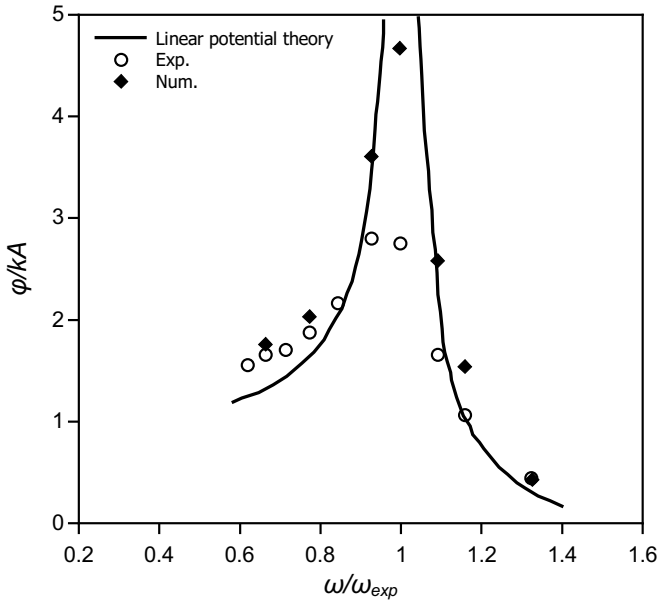


Figure 5-23: Magnification factors for roll motion.

with the experimental data and the solutions from the linear potential flow theory (digitised from Jung et al. (2006)). It can be seen that the present numerical results are overall more energetic than those of the experiment, which is consistent with the smaller damping effects as discussed in the free decay test.

In this test case, it is found that the flow motion around the corners of the box has different effects on the roll motion of the box for different wave frequencies, due to different mechanisms of vortex generation and evolution. Figure 5-24 and Figure 5-25 present the computed velocity and vorticity fields for wave frequencies $\omega = 6.76$ rad/s and $\omega = 5.24$ rad/s, respectively. While the former is a resonant case ($\omega/\omega_{exp} \approx 1.0$), the latter is a longer wave case ($\omega/\omega_{exp} \approx 0.77$). It can be

seen from Figure 5-24 (the resonant case, $\omega/\omega_{exp} \approx 1.0$) that the development of the vortex near the bottom corners mainly depends on the roll motion of the box. When the box reaches its maximum angle in the clockwise direction and rotates back (Figure 5-24(a) to (d)), the previously generated positive vortices, around the corners of the box on both the seaward and leeward sides, are quickly damped out and negative vortices are formed. This procedure reverses as the box starts rolling in clockwise direction (Figure 5-24(d) to (a)). This vortex generation and evolution causes the viscous damping effect that reduces the roll motion of the box.

In contrast, the longer wave case $\omega = 5.24$ rad/s ($\omega/\omega_{exp} \approx 0.77$), presented in Figure 5-25, reveals a different mechanism of vortex development around the corners of the box, due to the fact that the wave motion is faster than the box motion in this case. In particular, the vortex generation on the seaward side (left panels in each sub-figure) is discussed here, only due to its relatively strong vorticity in comparison to that of the leeward side. It can be seen that as the box reaches the maximum angle of clockwise motion, a negative vortex is generated ahead of the roll motion of the box (Figure 5-25(a)), which is the opposite of that in the resonant wave case (see Figure 5-24(a)). Furthermore, as the box rotates in a counterclockwise direction (Figure 5-25(a) to (d)), this negative vortex is quickly reversed, and a positive vortex is formed and stays ahead of the roll motion of the box (Figure 5-25(d)), which is also contrary to that shown in Figure 5-24(d). It seems that on the seaward side the vortex is mainly generated by the wave motion rather than the box motion, and this vortex generation mechanism, as noted by Jung et al. (2006), actually leads to an amplification of the roll motion rather than a damping of the motion.

Finally, it is worth noting that the present computational results in terms of the vortex development around the corner of the box are qualitatively consistent with the experimental and numerical results presented in Jung et al. (2006) and Jung et al. (2013). This indicates that the present "full particle" PIC model is capable of dealing with complex problems of viscous flows interacting with structures with angled corners.

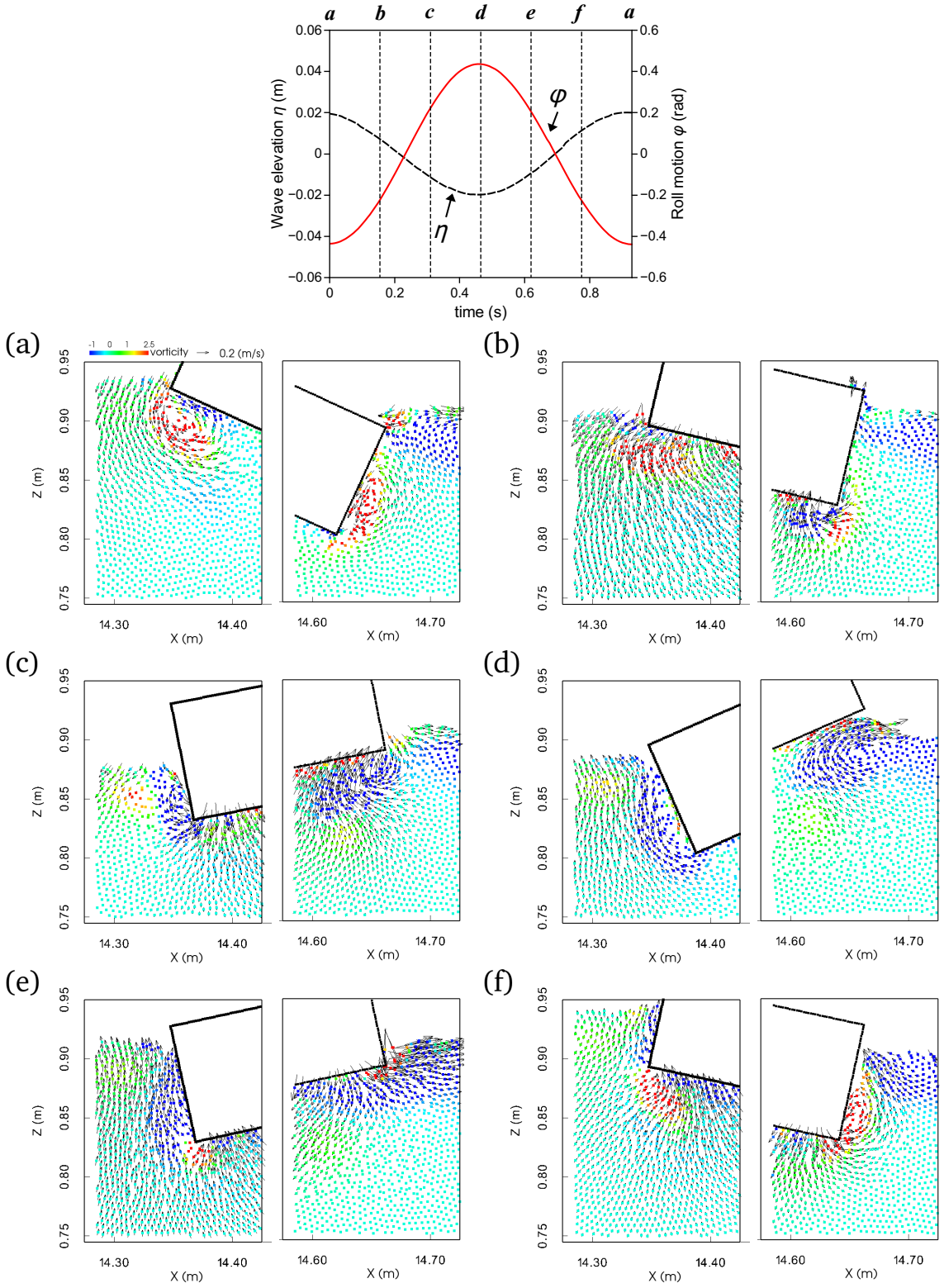


Figure 5-24: Measurements for $\omega = 6.76$ rad/s ($\omega/\omega_{exp} \approx 1.0$) wave. The top panel shows free-surface elevation of the incident wave in the absence of the box and the roll motion of the box with phases a–f corresponding to each of the velocity and vorticity snapshots. The vorticity is normalised by 3.132 s^{-1}.

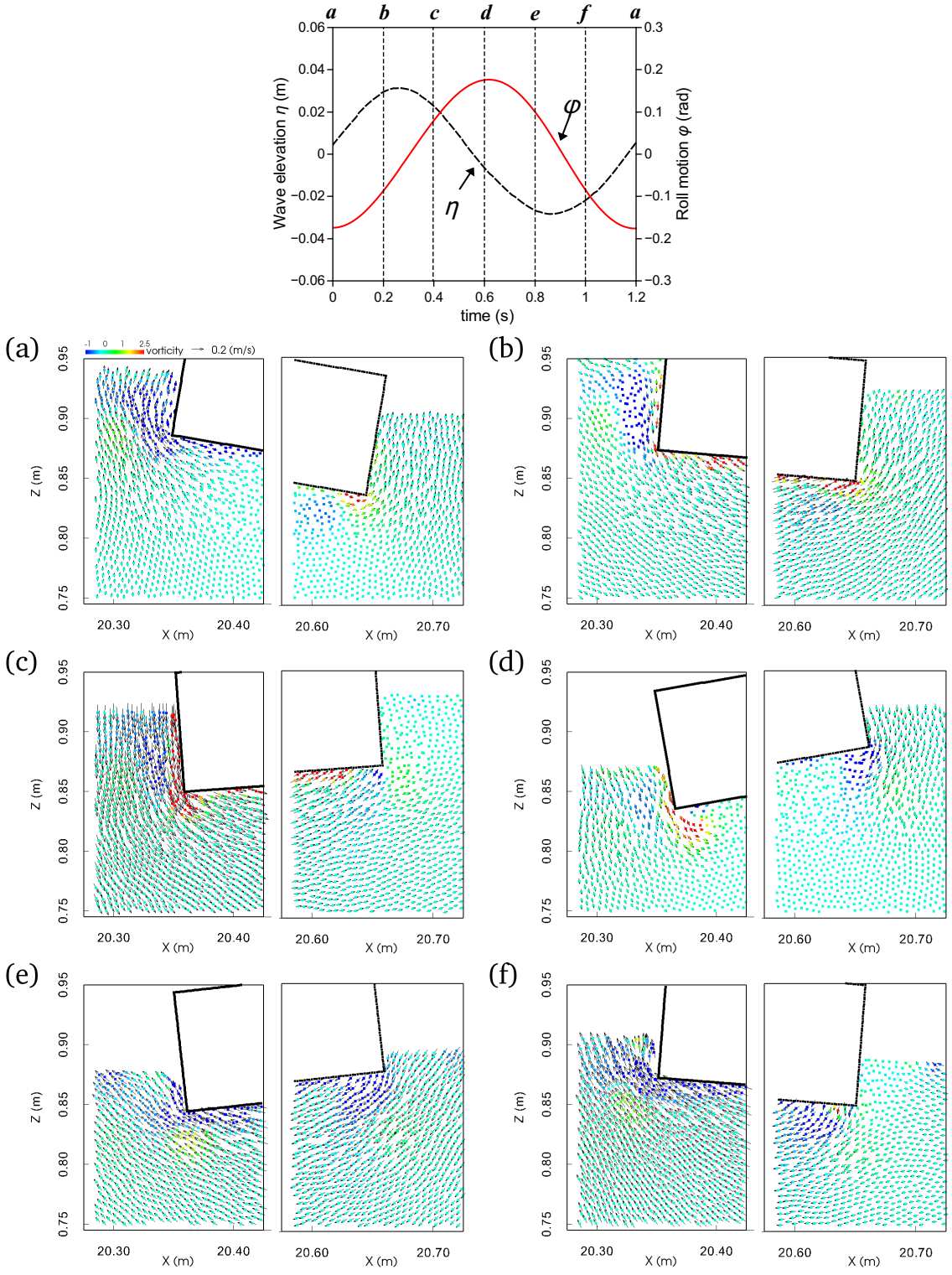


Figure 5-25: Measurements for $\omega = 5.24$ rad/s ($\omega/\omega_{exp} \approx 0.77$) wave. The top panel shows free-surface elevation of the incident wave in the absence of the box and the roll motion of the box with phases a–f corresponding to each of the velocity and vorticity snapshots. The vorticity is normalised by 3.132 s^{-1}.

5.6.5 Floating body under extreme wave conditions

This test case concerns the simulation of the coupled heave and roll motion of a floating body under both regular and extreme focused wave conditions. This case was previously investigated by Zhao and Hu (2012) where experimental data as well as the numerical results from Constrained Interpolation Profile (CIP) method are available for comparison. The experiments were carried out in a wave tank at the Research Institute for Applied Mechanics (RIAM) Kyushu University, Japan. The wave tank was 18 m long, 0.3 m wide and 0.7 m high and was equipped with a plunger-type wave generator and a wave absorbing device at each end. The water depth was fixed at $h = 0.4$ m throughout the experiments. Figure 5-26 shows a schematic of the experimental setup. The floating body was a symmetrical rectangular box with a deckhouse on the top, and major parameters of this structure are given in Table 5.3. The floating body was connected with a heaving rod through the rotational joint which was initially placed at the free surface of calm water. The heaving rod can move smoothly through the slider mechanisms fixed on guide rails on the top of the tank such that the floating body can move in heave and roll motion, with surge motion being restricted. The mass of the floating body and heaving rod were $m_1 = 14.5$ kg and $m_2 = 0.276$ kg, respectively. Wave gauges were placed upstream of the floating body to measure free-surface elevation and a pressure sensor was installed on the seaward side wall of the deckhouse, at a height of 0.01 m above the deck (see Figure 5-26), to monitor impact pressures.

For this test case, both regular waves and focused waves were generated in the experiment. The regular wave period was chosen to be $T_w = 1$ s and the wave amplitude

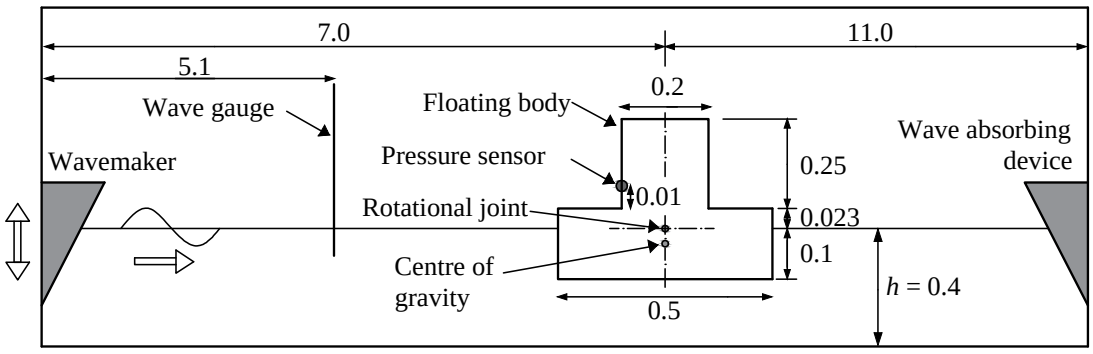


Figure 5-26: Sketch of experimental set-up in Zhao and Hu (2012); all dimensions in (m).

Table 5.3: Major parameters of the floating structure.

Item	Value (m)
Length	0.5
Breadth	0.29
Draft	0.1
Centre of gravity (from the bottom)	0.0796
Gyration radius	0.1535

was $A = 0.031$ m. For the focused wave, the peak frequency was $f_p = 1.0$ Hz with frequency of wave components ranging from 0.6 to 1.6 Hz. The focused position, time and amplitude were $x_f = 7.0$ m, $t_f = 20$ s and $A_f = 0.06$ m, respectively. Note that the wave steepness for the regular wave was 0.133, and it was 0.258 for the focused wave using the wave number of the peak frequency component and the focused wave amplitude; these values imply that weakly and strongly nonlinear waves were used for this test case, respectively.

To save on CPU cost for the current numerical simulation, following Zhao and Hu (2012), the distance from the centre of floating structure to the downstream end of the tank was reduced to 7 m, with 6 m dedicated to the wave absorption zone. Both the regular and focused wave were generated using the piston-type wavemaker technique as discussed in Section 5.5. For the focused wave, 29 wave components were used and their amplitudes were computed using Equation 5.26. The grid size was $\Delta x = \Delta z = 0.00625$ m, resulting in a cell resolution of 2264×120 with about 574,000 fluid particles. The maximum Courant number was 0.4 for the time step control and it took approximately 7 h to complete a total 30 s of simulation for the focused wave case.

Figure 5-27 shows comparisons between experimental and numerical results for structure motion and free-surface elevation under the regular wave condition. The top sub-panel presents the predicted free-surface elevation measured at the location of the structure (in the absence of the structure). This time series from the numerical results is presented merely for completeness; no experimental data for this is available. It can be observed that slight nonlinear effects occur in the incident regular wave. The two sub-panels in the middle compare the heave and the roll motions of the structure, respectively. It is seen that the present model under predicts the averaged amplitude of steady state heave motion (by approximately 7.7%) but over-predicts the roll motion (by approximately 23.4%). These discrepancies are probably

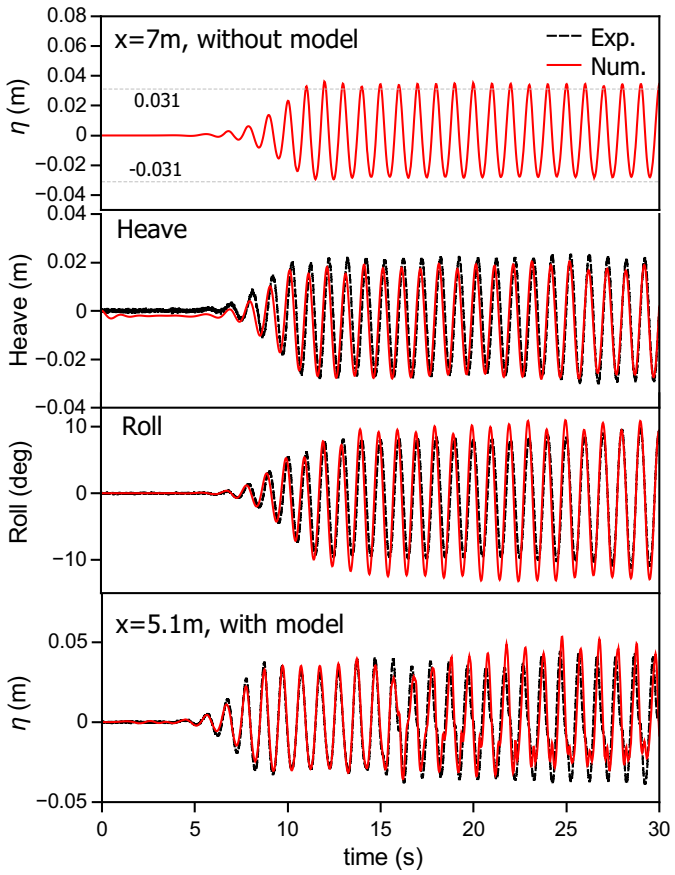


Figure 5-27: Comparison of the predicted and measured time histories for the regular wave case ($A = 0.031$ m).

because some wave-on-deck effects that occurred in the experiments (c.f. Figure 14 of Zhao and Hu (2012)) are not fully resolved by the numerical model, due to the limitation of 2D modelling and the inadequate resolution of computational grid. It is suggested in the work of Hu and Kashiwagi (2009), where the same experimental model is used, that the static and the dynamic effects of the green-water mass have important influences on the heave and the roll motions of this structure, respectively. Furthermore, mechanical friction effects present in the experiment (this is noted in Hu and Kashiwagi (2009)) are neglected in the present numerical modelling; this is also likely to contribute to the over prediction of the roll motion. The sub-panel at the bottom shows a comparison of the water surface elevation at $x = 5.1$ m (in the presence of the structure). The discrepancy between the free-surface elevations is clearly due to the complex wave refection that involves the structure motion and the wave-on-deck effects. Nevertheless, it may be worth noting that, in terms of this comparison of free-surface elevation, the present result is better than that of the CIP model (c.f. Fig. 13 of Zhao and Hu (2012)).

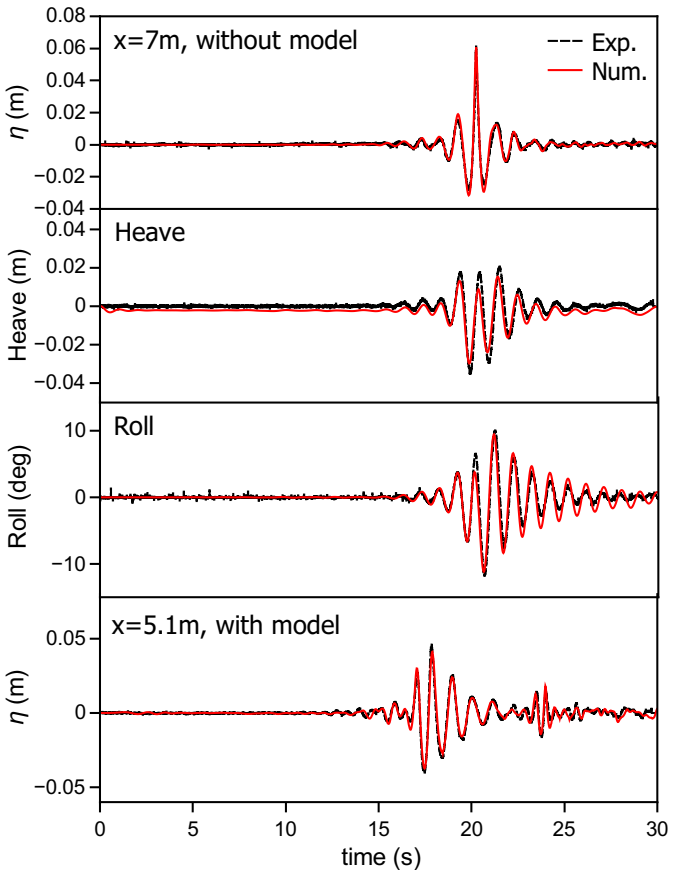


Figure 5-28: Comparison of the predicted and measured time histories for the focused wave case ($A_f = 0.06$ m).

For the focused wave case, the input of focused wave amplitude and time are slightly adjusted in the numerical model, in order to make the wave groups focus with $A = 0.06$ m at $x_f = 7$ m, where the floating body is placed. In the absence of the floating body, the numerical focused wave was first generated and the wave profile at the focused position is compared with the experimental data as shown in the top sub-panel of Figure 5-28. This configuration of focused wave generation was then used for the investigation of wave interaction with the floating body. Similar to the regular wave case, the other sub-panels of Figure 5-28 show the comparisons of results under the focused wave condition. It can be seen that the present model underestimates the nonlinear heave motion of the floating body but slightly overestimates the roll motion, which is consistent with the results discussed in the regular wave case. In terms of the water surface elevation at $x = 5.1$ m, it is seen that the numerical results agree well with the experimental data. In general, for the focused wave case, the numerical predictions by the current model are in good agreement with those of the experiment.

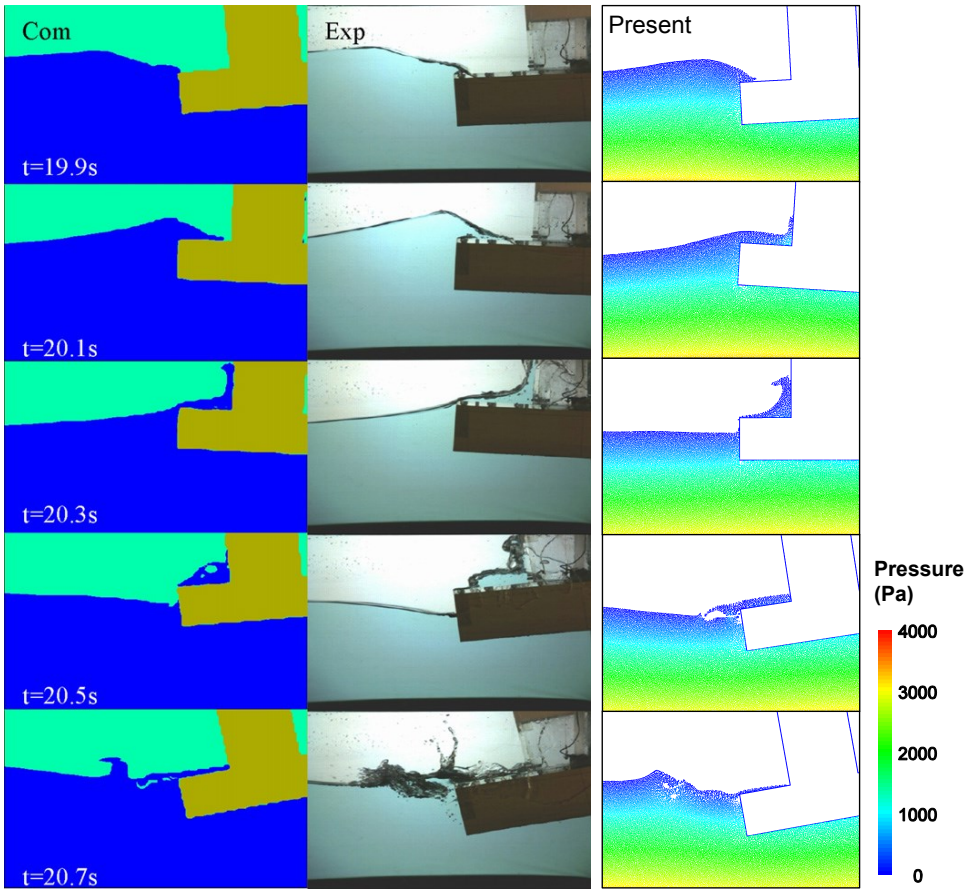


Figure 5-29: Comparison of snapshots between results of the CIP model (left), the experiment (middle), and the present "full particle" PIC model (right) under the focused wave condition ($A_f = 0.06$ m). The former two columns are reprinted from Zhao and Hu (2012), Copyright (2012), with permission from Elsevier.

Figure 5-29 shows a set of numerical snapshots at the same time instants with the experimental photographs for the focused wave case. We can see that the focused wave reaches the floating body and propagates onto the deck, leading to a flow impact on the deckhouse. The impact jet runs up and then overturns due to gravity, resulting in violent wave breaking in front of the structure. The present model roughly captures the major processes of this violent fluid-solid interaction when compared with the experiment. It is also noted that the present numerical predictions seem to be slightly earlier in time. The slightly earlier arrival of green water may be due to the discrepancy in the generated numerical and experimental focused waves. It was found that reproducing exactly the same focused wave as the experiment is very challenging.

The time history of fluid pressure on the pressure sensor (see Figure 5-26) is also

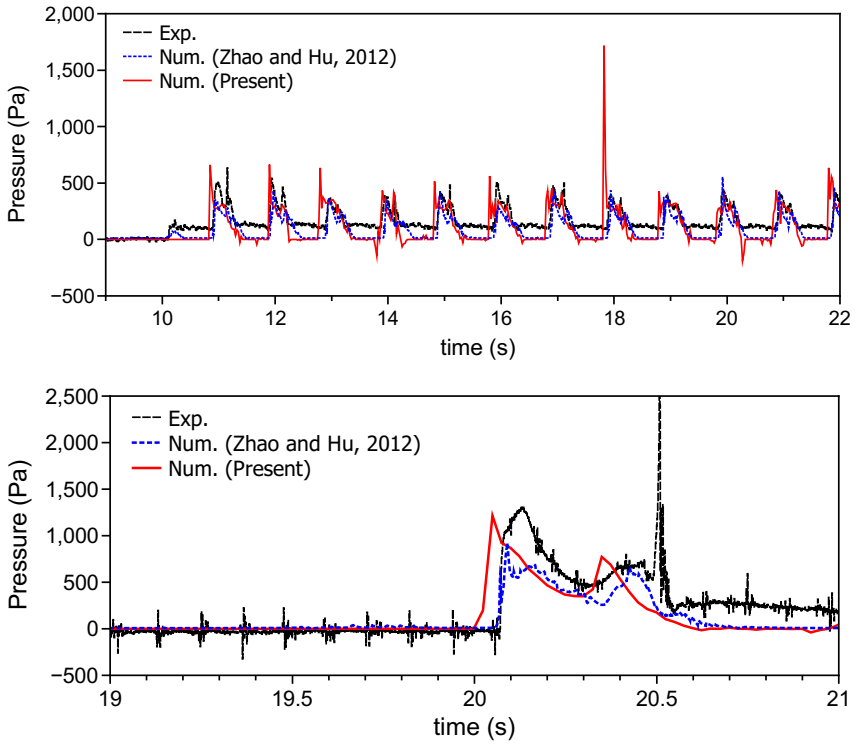


Figure 5-30: Comparison of the impact pressure under the regular wave (upper panel) and the focused wave (lower panel) conditions.

compared between experimental and numerical data, and this is shown in Figure 5-30, from which it is observed that the present model captures the variation of the pressures well for both the regular and the focused wave conditions. In particular, peak pressures during the regular wave conditions are very well captured, although the experimental data show that during wave run-down there is a higher residual pressure, which may be due to the slower water drainage than that predicted by the numerical model. Also, the present results appear to be noisier than those of the CIP model. This is likely to be caused by the effects of the small cut-cells as discussed in Section 5.6.3.

5.7 Summary

This chapter presents, within the "full particle" PIC framework, a Cartesian cut cell based two-way strong fluid-structure coupling algorithm that allows for the simulation of arbitrarily configured (surface-piercing) floating bodies of single or multiple

DoF. This coupling scheme combines the cut cell approach detailed in Ng et al. (2009) with ideas of the strong fluid-structure coupling technique proposed in Batty et al. (2007). The key point of this coupling methodology is that, in the finite volume discretisation, the velocity integral along the structure surface is implicitly and purely represented by fluid pressure in cells immediately surrounding the structure, and this is eventually integrated into the solution of the PPE. This strong coupling scheme makes the current "full particle" PIC model much more stable and able to handle violent fluid-structure interactions than a weak coupling scheme. It should also be straightforward to implement this scheme within other projection based incompressible flow solvers, and/or extend it to three spatial dimensions.

A NWT has been established within the "full particle" PIC model. A piston-type wavemaker has been developed for wave generation. This is achieved by using the fluid-structure coupling algorithm proposed in this chapter. Also, a relaxation method is employed for the wave absorption at the downstream end of the NWT. Both the wave generation and absorption techniques work well as demonstrated in the test cases presented in this chapter. The waves studied in this chapter include regular and focused waves. A solitary wave generation technique is also described and this will be tested in the following chapter in three spatial dimensions.

The fluid-structure coupling algorithm, as well as the present PIC model, are validated by simulating wave interaction with fixed, prescribed motion and freely movable structures of different shapes in two spatial dimensions. Comparisons between the present numerical results and those from either other numerical methods or physical experiments show reasonable agreement. It is demonstrated that the present model is capable of handling, in a straightforward manner, violent free-surface deformations such as wave overtopping (see Section 5.6.1) and wave breaking and slamming (see Section 5.6.5). The present model also deals with both nonlinear effects (see Section 5.6.2 and Section 5.6.5) and viscous effects (see Section 5.6.4) with reasonable accuracy. Moreover, in Section 5.6.1 and Section 5.6.3, it is shown by rough comparisons that the present "full particle" PIC model seems to be much more efficient than the particle-based PFEM model of Oliveira et al. (2012) and the SPH model of Westphalen et al. (2014), in terms of CPU cost.

However, it is also noted that the accuracy and stability of fluid pressure computation in the cut-cells requires further improvement (see Section 5.6.3 and Section 5.6.5),

such as using the cell-merging technique proposed in Kirkpatrick et al. (2003) to eliminate small cut-cells. Also, to mitigate the demanding memory storage when a very fine computational grid is required, an adaptive grid, or alternatively a distributed memory based parallelisation approach, could be developed. The latter approach is implemented for the 3D modelling introduced in the following chapter.

CHAPTER 6

A 3D PARALLEL PIC MODEL FOR WAVE INTERACTION WITH VERTICAL CYLINDERS

In this chapter, the "full particle" PIC model, incorporating the Cartesian cut cell based scheme proposed in chapter 5 for fluid-structure interaction, is extended to three spatial dimensions and parallelised using the MPI approach. Validation of the 3D parallel model concentrates on test cases involving multiple types of water waves, such as regular, focused and solitary waves, interacting with vertical cylinders in several spatial configurations. The results are compared with laboratory data and numerical results from state-of-the-art VOF based Eulerian solvers such as those from the OpenFOAM® suite. It is shown that the present parallel model is capable of simulating highly nonlinear water waves and the interaction of such waves with vertical cylinders in three spatial dimensions with a CPU efficiency slightly less than Eulerian solvers. Moreover, the innovative use of particles to track the free surface and solve the nonlinear advection term, akin to meshless Lagrangian solvers, gives the model a particular flexibility in handling complex, full 3D water-wave scenarios involving large free-surface deformations.

6.1 Introduction

Extending the present PIC model to three spatial dimensions comes with additional requirements for computational resources due to the highly demanding memory storage, as the double grid system is employed. To mitigate this problem, one possible solution is to use a dynamic adaptive grid and augment it with the particle merging/splitting technique such as that proposed in Hong (2009). However, changing the grid system using these techniques brings difficulties from not only the adaptive grid

itself but also other complications such as the free surface handling. An alternative, and potentially simpler, option is to parallelise the PIC model using the distributed memory based MPI approach (see `www.open-mpi.org`), such that large-scale problems (e.g. those require a large number of grid cells and particles) can be simulated. This thesis chooses the latter approach to develop the "full particle" PIC model used in chapter 5 for applications in three spatial dimensions.

The validation tests are based on wave interaction with vertical cylindrical structures, which are widely employed in coastal and offshore engineering. Examples include the design of oil platforms, offshore wind turbine foundations and piled wharfs (Zhu and Moule, 1996). As a consequence it is important to understand the interaction between waves and single or multiple cylinders, in terms of free-surface elevations around the cylinders, wave run-up and wave loading. Numerous studies, both numerical and experimental, have been carried out on this topic. For example, Chen et al. (2014b) employed the OpenFOAM® model to study the nonlinear effects of both regular and focused wave interaction with a single cylinder. This case was experimentally studied by Zang et al. (2010). Mo and Liu (2009) investigated non-breaking solitary wave interaction with a single cylinder and a group of three cylinders through a VOF based finite volume numerical solver and also physical experiments. Lara et al. (2013) and Leschka and Oumeraci (2014) also investigated solitary wave interaction with three vertical cylinders based on OpenFOAM®; in these works, different configurations of the cylinders are studied. In addition to the above-mentioned Eulerian methods, Lagrangian methods have also been widely employed for investigating these topics, such as the SPH method (see e.g. Lind et al. (2016)) and the meshless local Petrov-Galerkin method based on Rankine source solution (MLPG_R) (Zhou, 2010).

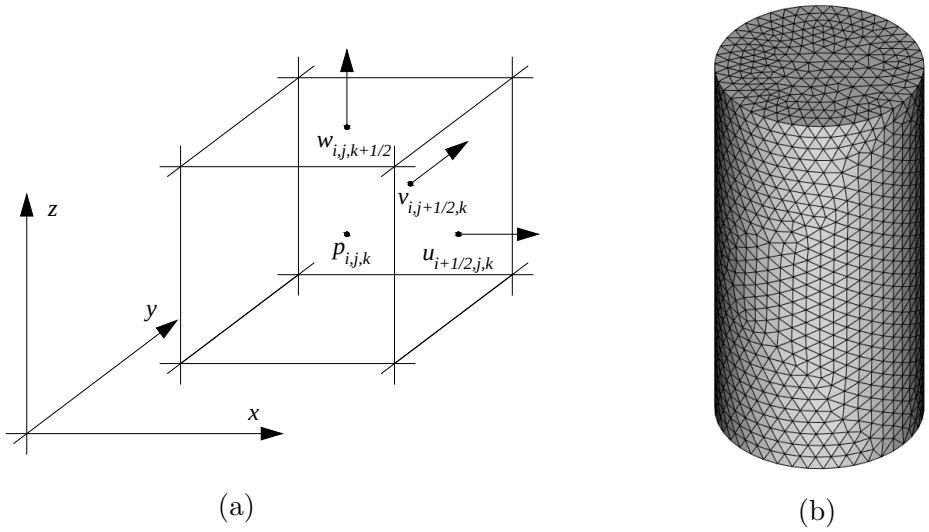
This chapter aims to contribute to further understanding the interaction between different waves and single or multiple vertical cylinders, in terms of free-surface motion, wave run-up and wave loading. More importantly, in order to demonstrate the capability of the 3D parallel model, attention has been focused on the comparison of results between the present model and state-of-the-art VOF-based Eulerian models such as OpenFOAM®.

6.2 The 3D model and parallelisation

6.2.1 Setup of the 3D model

The 3D model uses the approaches described in chapter 3 to solve the incompressible Newtonian NS equations for fluid flows, and employs the cut cell technique introduced in chapter 5 for modelling fluid-structure interactions.

The 3D computational domain is discretised by cubic cells and the interface of rigid structures are discretised using triangular elements; Figure 6-1 shows examples for both discretisations. In this study, the triangular elements are generated using the open-source tool SALOME (@CEA/DEN et al., 2014), with a characteristic area of approximately $(\Delta x)^2/55$.



(a) (b)

Figure 6-1: Discretisation of the computational domain: (a) domain grid; (b) structure grid.

For the 3D model, the fraction open-to-water, E, utilised in Equation 5.8 changes from line segments to face segments. They are computed by assuming that the structure interface in the cubic cells is just a plane that cuts the cell face into only trapezoidal or triangular segments as shown in Figure 6-2. To calculate the area of such a face segment, the length of each grid line that is not occupied by structures (e.g. e_1 in Figure 6-2) is computed, using the approach introduced in Section 5.3.3. Also, for numerical stability it is implemented in the 3D model that when the area

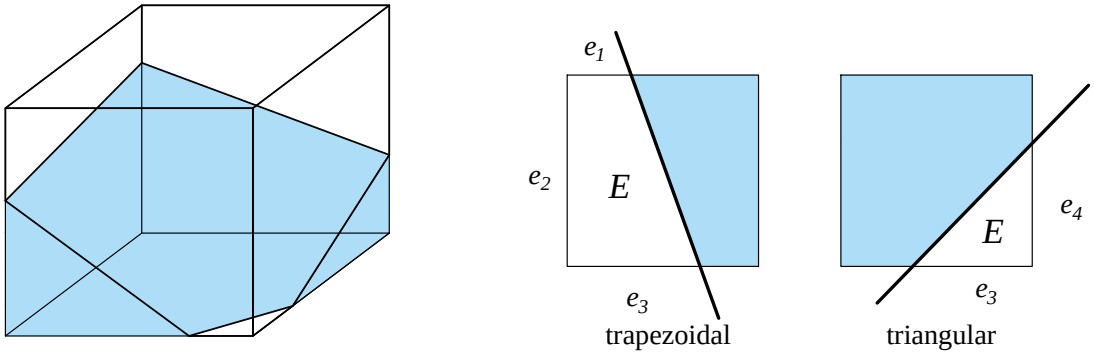


Figure 6-2: Open-to-water fractions (E) of grid faces cut by structures (shaded area) in three dimensions.

of a face segment open-to-water is less than $0.1(\Delta x)^2$, the area of such face segment is set to zero. It is nevertheless noted that the above method may not be able to capture some sharp features of structures. An alternative option could be to discretise each grid face into a number of small faces and then use the method discussed in Section 4.3 to compute the fractions, but this is expected to be very costly in terms of CPU time.

6.2.2 Parallelisation of the 3D model

This section introduces the major components of the parallelisation of the current "full particle" PIC model using the MPI approach.

Domain decomposition

By using the MPI approach, the computational domain is first divided into a number of sub-domains, which are then solved using different processors. Figure 6-3 shows a schematic of the sub-domains. Information transfer (e.g. particle and grid properties) is conducted between processors using the one layer ghost cells at the boundaries of each sub-domain. Particles entering the ghost cells after the advection step are removed from the present sub-domain and moved to the adjacent sub-domains, along with all the quantities they carry. Note that in certain instances, for example, when computing the fourth-order WENO scheme for grid-to-particle interpolation

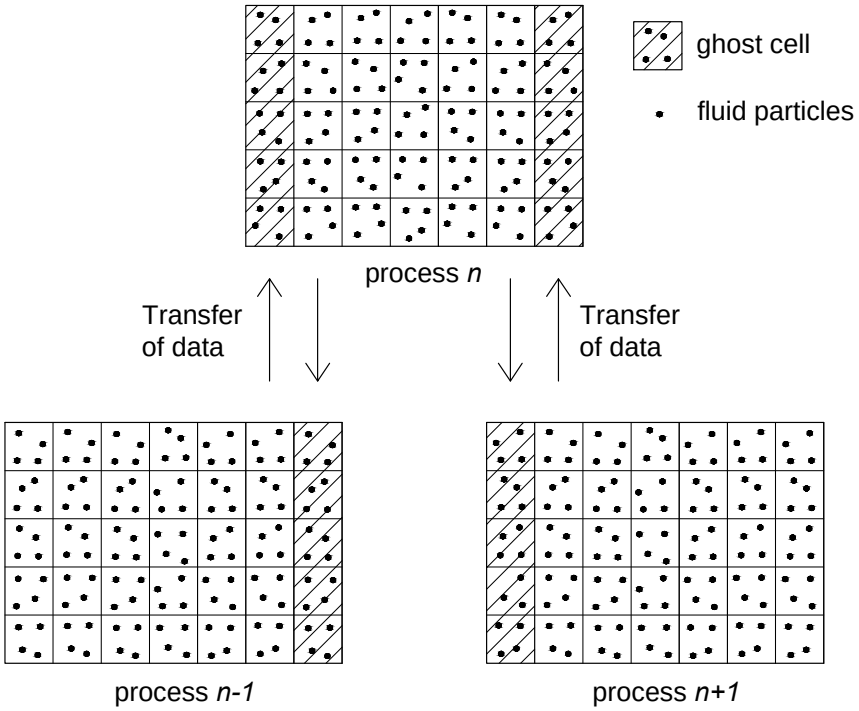


Figure 6-3: Domain decomposition and data transfer via ghost cells.

(see Section 3.2.2), one additional layer of ghost cells is used, as four-point stencils are required.

The parallel BCG solver for PPE

The serial BCG solver of Press et al. (1992) is parallelised using the MPI approach for the present model. Given a linear system of equations $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ with preconditioner $\tilde{\mathbf{A}}$ (the diagonal part of $\mathbf{A}$), Figure 6-4 presents the algorithm for the parallelised BCG solver. It can be seen that the BCG algorithm requires several Matrix-Vector multiplications and Vector-Vector products. This is where data communication is required, as the matrix $\mathbf{A}$ is divided into the sub-domains. In Figure 6-4, this is marked by bullet points.

Structure handling and relevant load balancing

In the present parallel model, the structure data are stored identically on each processor, rather than being split across different processors. As such, in future develop-

$\mathbf{x}_1$ = Initial guess

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A} \cdot \mathbf{x}_1$

- *Update* $\mathbf{r}_1$ *in ghost cells using MPI_SENDRECV*

$\bar{\mathbf{r}}_1 = \mathbf{r}_1$

Set $\mathbf{p}_1 = \mathbf{r}_1$ and $\bar{\mathbf{p}}_1 = \bar{\mathbf{r}}_1$

Repeat k = 1,2...

Set $\tilde{\mathbf{A}} \cdot \mathbf{z}_k = \mathbf{r}_k$ and $\tilde{\mathbf{A}}^T \cdot \bar{\mathbf{z}}_k = \bar{\mathbf{r}}_k$

- *Sum up* $\bar{\mathbf{r}}_k \cdot \mathbf{z}_k$ *in all processes using MPI_ALLREDUCE(MPI_SUM)*
- *Update the products of* $\mathbf{A} \cdot \mathbf{p}_k$ *in ghost cells using MPI_SENDRECV*
- *Sum up* $\bar{\mathbf{p}}_k \cdot \mathbf{A} \cdot \mathbf{p}_k$ *in all processes using MPI_ALLREDUCE(MPI_SUM)*

$$\alpha_k = \frac{\bar{\mathbf{r}}_k \cdot \mathbf{z}_k}{\bar{\mathbf{p}}_k \cdot \mathbf{A} \cdot \mathbf{p}_k}$$

$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \cdot \mathbf{p}_k$

- *Update the products of* $\mathbf{A}^T \cdot \bar{\mathbf{p}}_k$ *in ghost cells using MPI_SENDRECV*

$\bar{\mathbf{r}}_{k+1} = \bar{\mathbf{r}}_k - \alpha_k \mathbf{A}^T \cdot \bar{\mathbf{p}}_k$

- *Sum up* $\bar{\mathbf{r}}_{k+1} \cdot \mathbf{z}_{k+1}$ *in all processes using MPI_ALLREDUCE(MPI_SUM)*
- *Sum up* $\bar{\mathbf{r}}_k \cdot \mathbf{z}_k$ *in all processes using MPI_ALLREDUCE(MPI_SUM)*

$$\beta_k = \frac{\bar{\mathbf{r}}_{k+1} \cdot \mathbf{z}_{k+1}}{\bar{\mathbf{r}}_k \cdot \mathbf{z}_k}$$

$\mathbf{p}_{k+1} = \mathbf{z}_k + \beta_k \mathbf{p}_k$

$\bar{\mathbf{p}}_{k+1} = \bar{\mathbf{z}}_k + \beta_k \bar{\mathbf{p}}_k$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$

- *Call MPI_BARRIER to keep all processes at the same pace*

Until stopping criteria is reached

Figure 6-4: An algorithm of BCG solver (Press et al., 1992) parallelised using the MPI approach. The bullet points indicate the intervention of parallelisation.

ment if structures such as floating bodies move across processors, the complex data communication is eliminated, which can greatly simplify the code implementation.

Moreover, the structure-induced computational loads (e.g. the computation of fractions open-to-water as mentioned above) on an individual processor can significantly lower the overall efficiency of the parallel computing. This is because, in most cases, only a few sub-domains are occupied by structures, where the structure-induced computational loads can waste the CPU time of other processors as they are kept waiting. To mitigate this problem, a general algorithm has been developed to assign local computational loads from some processors to all available processors. This

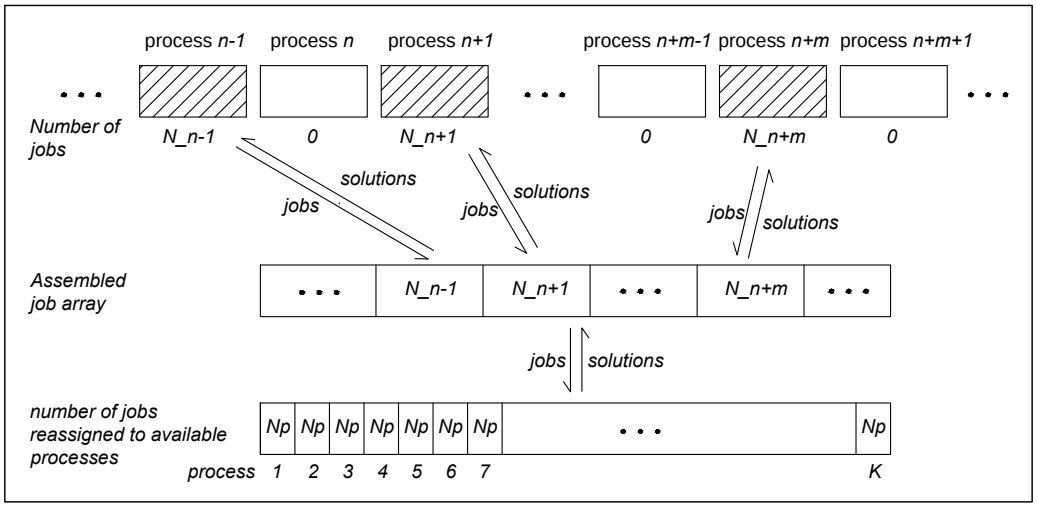


Figure 6-5: An algorithm for assigning computational loads from local processors to all available processors.

algorithm is illustrated in Figure 6-5. Specifically, once the structure-induced local computational loads are detected, they are assembled into an array of jobs, which includes the specific computational task and the linking information (i.e. the source processor of the computational task). These jobs are then reassigned evenly amongst all available processors to be conducted. Once the jobs are finished, the solutions will be collected and sent back to their source processors using the linking information.

Test of scalability

The performance of parallelisation in the present model is tested through the simulation of solitary wave propagation in the NWT as described in Section 6.3.2. For the performance test here, a total number of 22.33 million particles and 7.78 million cells were used (note that in this test case while the water depth was 0.75 m, the height of the NWT was 2 m). The simulation was run on the Bath HPCS, which is based on the Intel Xeon E5-2650v2 IvybridgeV2 processor. Figure 6-6 presents the trends of speedup (S_p) and efficiency (E_p), where

$$S_p = \frac{T_k}{T_p} \quad \text{and} \quad E_p = \frac{S_p}{p/k}\,; \tag{6.1}$$

T_p and T_k represent the CPU time, with p and k being the number of cores. Note that, ideally, in the above equations k should be 1, i.e. the serial computation. For the present PIC based solver, it however would not be practical as the memory

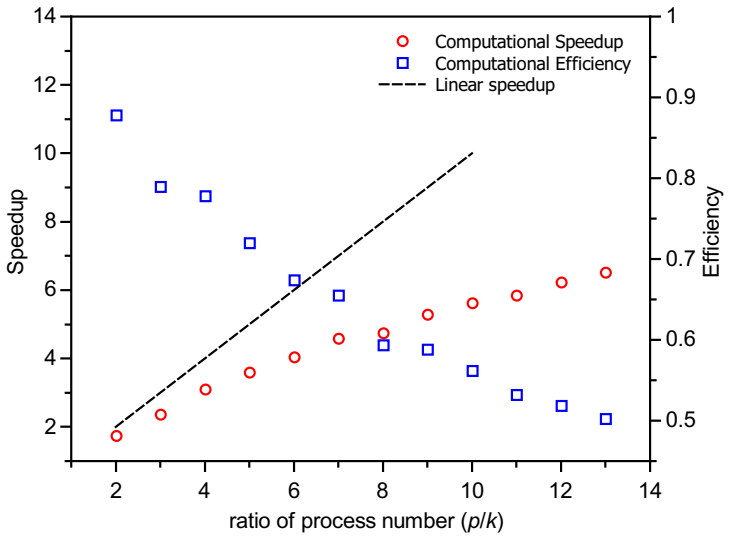


Figure 6-6: Speedup and efficiency trends. $k = 16$.

requirement due to the double grid system for such computation is rather high for a single processor. Thus, k was chosen to be 16 in the present computation. From Figure 6-6, it can be seen that the efficiency drops generally from around 0.9 to 0.5 as the number of processors increases from 32 ($p/k = 2$) to 208 ($p/k = 13$). This implies that the computational costs for non-parallel operations such as data communication become more prominent as a larger number of processors are used. The decrease in computational efficiency is also reflected in the sub-linear speed-up trend. Nevertheless, it can be seen that when the number of processors is relatively small, e.g. around $p/k = 3$, the speedup is close to linear.

6.3 Test cases

In this section, a number of test cases are presented in which the 3D parallel "full particle" PIC model is validated against experimental data as well as results from other state-of-the-art numerical models. Attention has been focused on the validation of the model for water wave interaction with vertical cylindrical structures. Two different test cases are selected: i) regular and focused wave interaction with a single cylinder (Chen et al., 2014b); ii) solitary wave interaction with single and multiple cylinders (Mo and Liu, 2009). All the test cases were run on the HPCS at the University of Bath, UK.

6.3.1 Wave interaction with a single vertical cylinder

Model setup

This test case concerns regular and focused wave interaction with a single cylinder, typically encountered in offshore wind turbine foundations. The test case was experimentally studied by Zang et al. (2010) and numerically simulated using OpenFOAM® by Chen et al. (2014b). The experiment was carried out in DHI, Denmark in a shallow water basin (35 m $\times$ 25 m), where waves are generated with a segmented piston paddle array installed at one end of the basin. The water depth, h, was set to 0.505 m. The cylinder, with a diameter of 0.25 m, was placed in the middle of the basin, with the distance between the upstream stagnation point of the cylinder and the wavemaker being 7.52 m. The main points of interest of this test case are the wave elevation around the cylinder, wave run-up and the horizontal wave force on the cylinder.

Four different wave conditions were investigated, and the parameters for these waves are given in Table 6.1, where "R" represents regular wave and "F1", "F2" and "F3" represent focused wave cases; T is the wave period; f is the wave frequency; A represents the wave amplitude; k is the wave number; a is the radius of the cylinder. Two different slenderness values (ka) and two different wave steepnesses (kA) are used for the focused waves. Note that, for the focused waves, the parameters belong to the peak frequency wave component. Also, for the wave conditions used here, the Keulegan-Carpenter number, $KC = \pi A/a$, is approximately 3.0 for the largest wave amplitude, although the validity of the KC number remains to be investigated for focused waves. This indicates that the studied cases are in an inertia-dominated regime (Paulsen et al., 2014). Thus, the dominant term in the wave loading on the cylinder is the inertia force. This is also noted by Chen et al. (2014b) where the laminar flow model of OpenFOAM-2.1.0 is used; their numerical results are used for comparisons for this test case.

Figure 6-7 shows a schematic of the setup of the NWT. Compared to the physical wave basin, the NWT is reduced to be 20.2 m long, 4 m wide and 0.9 m high in order to save on CPU cost. The distance between the cylinder and the wavemaker was set to be the same as that used in the experiment. The length of the absorption zone

Table 6.1: Experimental wave parameters used in this study.

Test conditions	R	F1	F2	F3
T (s)	1.22	1.22	1.63	1.22
f (Hz)	0.82	0.82	0.61	0.82
A (m)	0.07	0.07	0.12	0.11
ka	0.37	0.37	0.25	0.37
kA	0.2	0.2	0.2	0.3
kh	1.39	1.39	0.86	1.39

(D_L) at the end of the basin was set to 4.7 m and 8.7 m for the regular and focused wave cases, respectively. The time histories of wave elevation at two locations were measured: the first wave probe (WP1) was placed at 0.77 m from the wavemaker; the second wave probe (WP2) was located 0.02 m in front of the cylinder.

Wave generation and absorption in the numerical modelling use the techniques described in Section 5.5. For the focused wave, 60 wave components were used and their amplitudes were computed using Equation 5.25.

A grid refinement study on the focused wave generation for F2, in the absence of the cylinder, was used to determine the grid size. Three different grid sizes were chosen: $\Delta x = \Delta y = \Delta z = 0.01$ m ($A/12$; the fine grid), 0.02 m ($A/6$; the moderate grid) and 0.04 m ($A/3$; the coarse grid). Here the width of the NWT was further reduced to 0.4 m to speed up the test, as wave profiles in the spanwise direction are

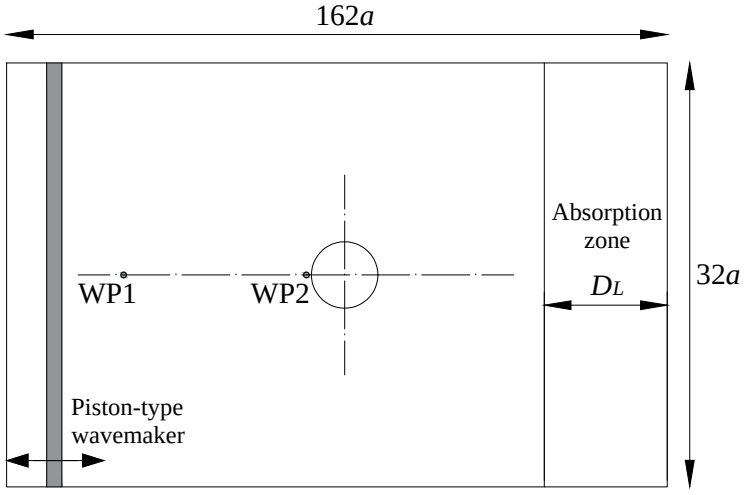


Figure 6-7: Schematic (top view) showing the setup of the NWT. a is the radius of the cylinder.

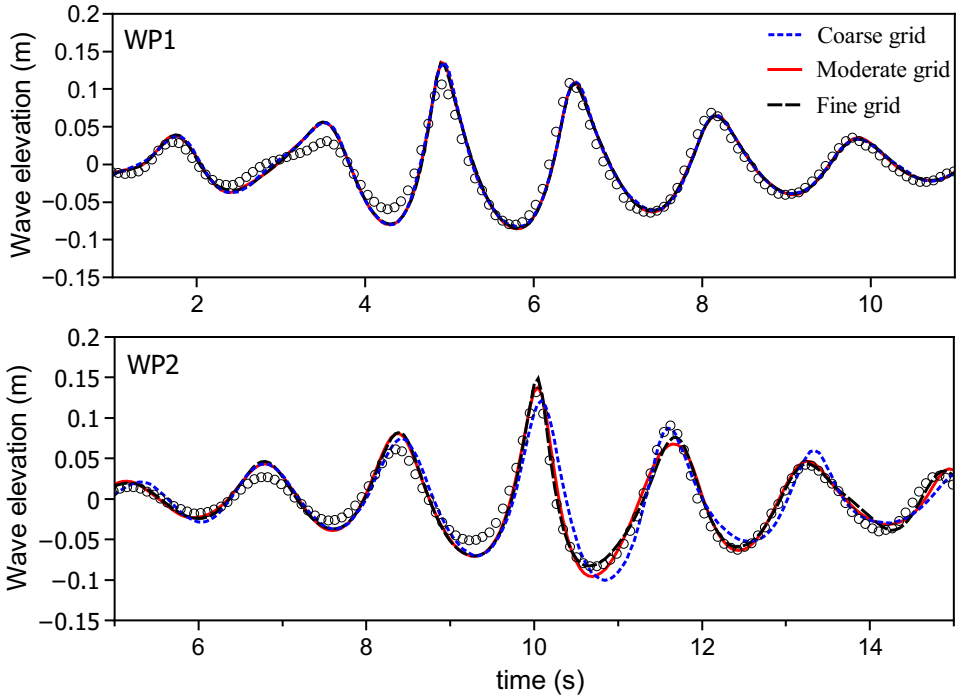


Figure 6-8: Grid refinement study of focused wave generation for F2. The circles represent the experimental data.

constant. Figure 6-8 plots wave profiles of the generated focused wave F2 for the three grid configurations at both WP1 and WP2. The results are also compared with the experimental data (the circles). It can be seen that the predicted wave profiles at WP1 are almost indistinguishable, suggesting that the wave characteristics are generated independently from the grid refinement. In addition, the predicted wave profiles match well with the experiment, indicating that the waves are generated correctly. At WP2, however, the wave profile produced by a coarser grid tends to have a larger discrepancy from the experimental data when compared with that of the fine grid; this is caused by the faster accumulation of dispersion and diffusion errors which result from a coarser grid over time as the wave propagates further inside the NWT. Nevertheless, the model is nearly convergent when using the moderate grid (0.02 m; $A/6$), and the result matches well with the experimental data. Considering the increased memory requirement of using the fine grid, the moderate grid is thus selected, resulting in approximately 9.2 million cells and 40.12 million particles in the presence of the cylinder. The Courant number used is 0.5, which is the same as that used in Chen et al. (2014b). The CPU time for the regular wave case is given below (Table 6.2) and compared with that of the OpenFOAM® model of Chen et al. (2014b).

Comparison of results for the regular wave case

Figure 6-9 shows snapshots of free-surface elevation around the cylinder. It is seen that as the crest of incident wave approaches the cylinder, it causes wave run-up and subsequent run-down of water on the cylinder. The scattered waves due to the cylinder are mostly of higher frequency, as the characteristic wavelengths are visibly shorter than that of the incident wave. Furthermore, the scattered waves interact with each other, as well as the incident wave, resulting in a steep "rooster tail" shaped wave at the lee side of the cylinder, which is very similar to the numerical result presented in Kim et al. (2006).

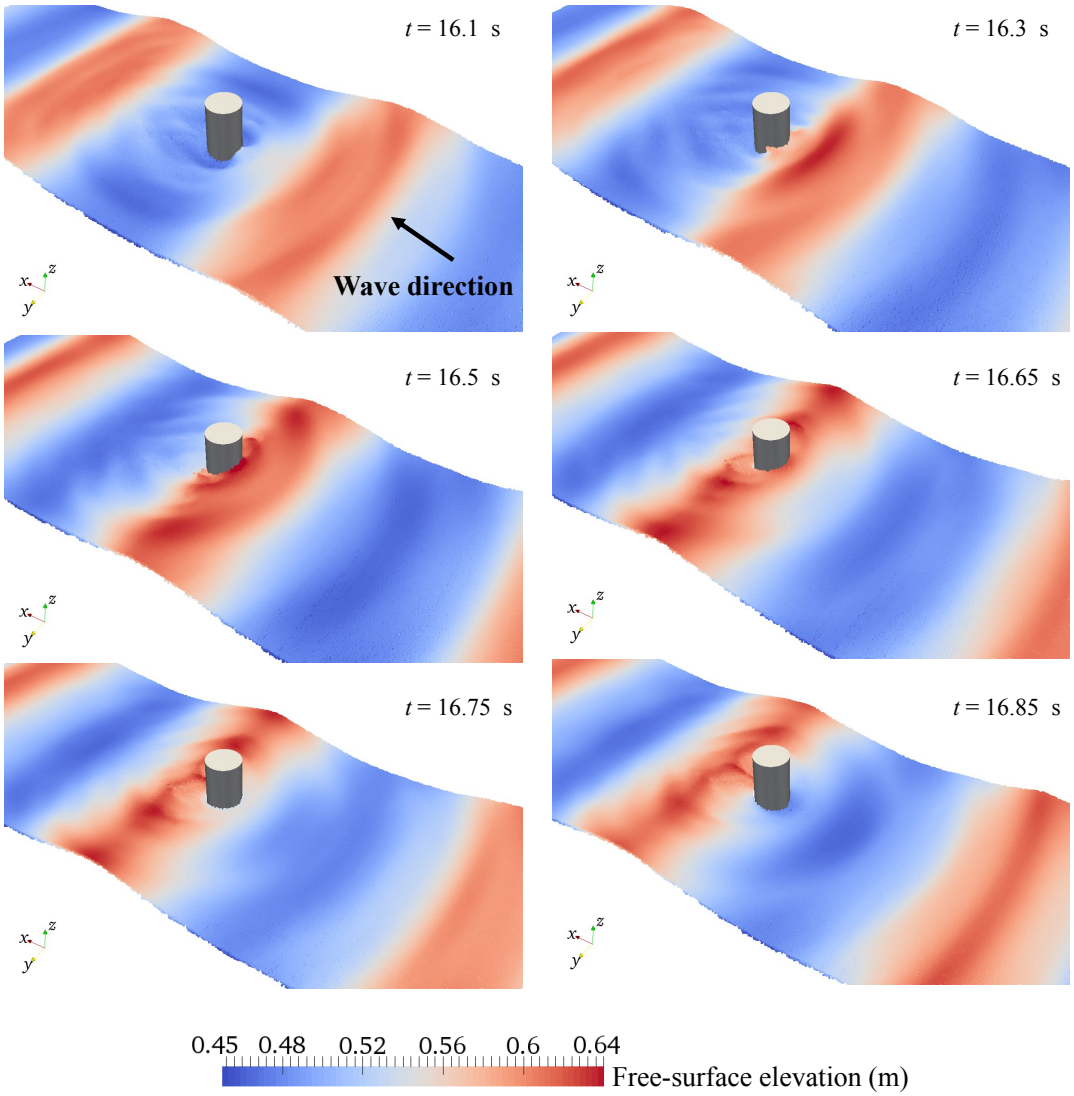


Figure 6-9: Snapshots of the numerical simulation for regular wave (R) interaction with a single cylinder.

Table 6.2: Comparison of CPU time between the present "full particle" PIC model and the OpenFOAM® model of Chen et al. (2014b) for the regular wave case.

Items	"full particle" PIC	OpenFOAM®
Cell number n (million)	9.20	8.33
Cores p	80	8
Simulated time t_s (T)	18.0	24.5
Total CPU time C_t (h)	12.16	126.20
Magnified CPU cost: $\frac{p \times C_t}{n \times t_s}$	5.87	4.95

Figure 6-10 presents quantitative comparisons between numerical predictions and experimental measurements for the steady state results, including wave elevations at WP1 and WP2, and the horizontal wave force on the cylinder. The corresponding amplitude spectra calculated by a fast Fourier transform (FFT) algorithm are also plotted for further comparison. It can be seen that, in general, results from both numerical models agree well with the experimental measurements, and the present "full particle" PIC model produces very similar results to those from the OpenFOAM® model of Chen et al. (2014b). In addition, from the comparisons of the spectra it is demonstrated that the present 3D model simulates the nonlinear effect of the wave dynamics well, although the high-order harmonics of the horizontal wave force are under-predicted.

Table 6.2 gives a comparison of the CPU time between the present model and the OpenFOAM® model. It is noted that this comparison is not strictly consistent in a few points: i) the present model is currently a single-phase solver, whilst the OpenFOAM® (interFoam) model is two-phase; ii) numerical schemes of free-surface capturing and solution of the advection term, and the grid configuration are different; iii) the simulations were run on different numbers of cores. Nevertheless, as can be seen from the magnified CPU cost, this rough comparison demonstrates that, for this type of modelling, the present PIC model is only slightly less efficient (in terms of CPU cost) than OpenFOAM®, which is one of the most widely used CFD models currently in the coastal engineering community (see e.g. Higuera et al. (2013b)).

Comparison of results for the focused wave cases

In order to generate similar focused waves as those used in the physical experiment, the experimental measurement of wave profiles at WP1 were used to calibrate the present numerical model; the wave profiles and the corresponding amplitude spectra

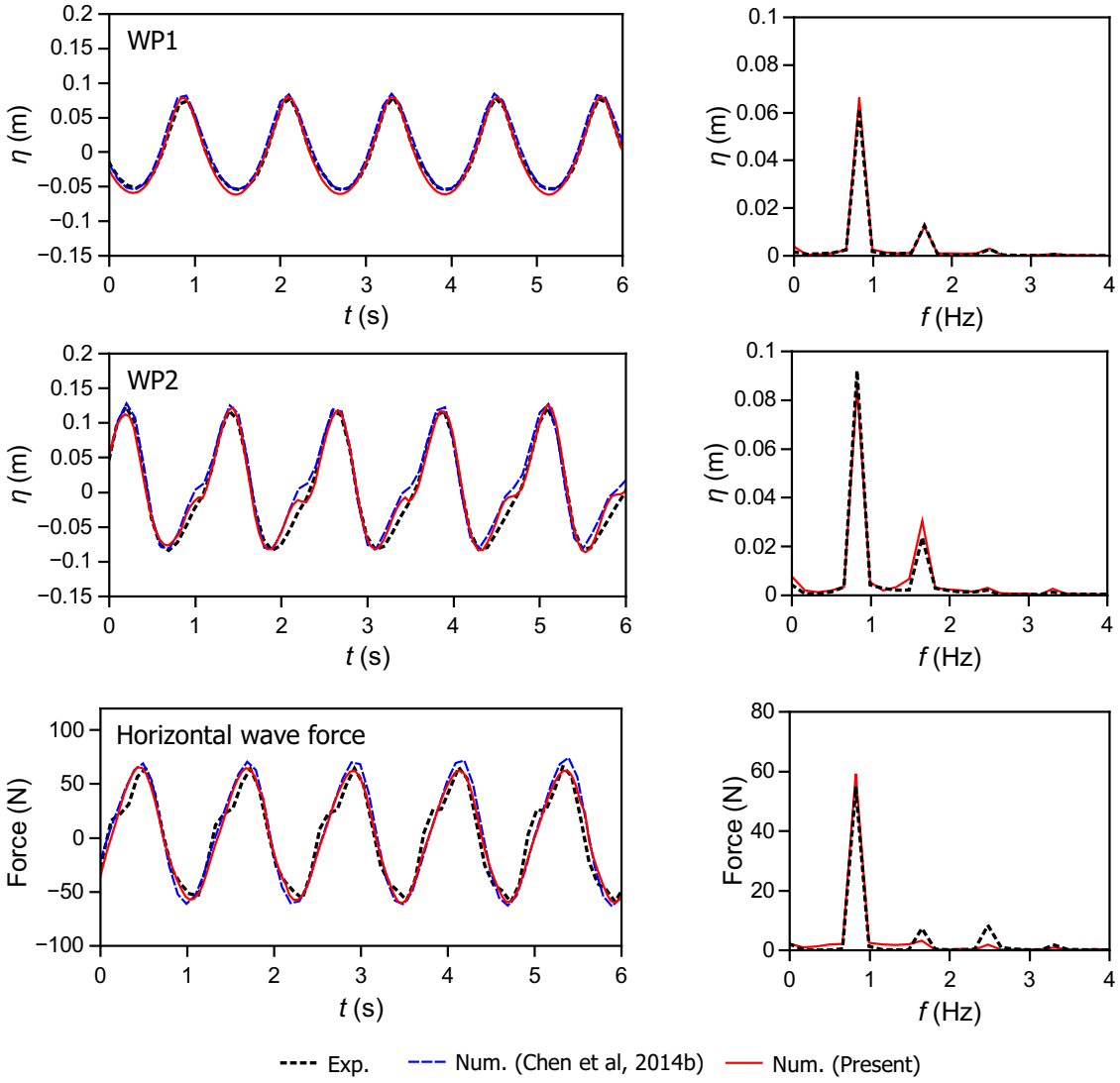


Figure 6-10: Comparisons of wave elevations at WP1 and WP2, the horizontal wave force and their corresponding spectra for the regular wave case ($ka = 0.37$; $kA = 0.2$).

are plotted in Figure 6-11. It can be seen that the numerical results of both wave elevations and wave energy distributions agree well with the experimental data for all the wave conditions. These ensure that the incident waves are well reproduced in the simulation to enable the investigation of wave-structure interaction.

Figure 6-12 and Figure 6-13 present comparisons of water surface elevations at WP2 and horizontal wave forces on the cylinder, respectively. Numerical results from the present model and the OpenFOAM® model of Chen et al. (2014b) (where only data for cases F1 and F2 are available) are plotted against the experimental data. It is seen that both numerical models achieve good agreement with the experiment. In

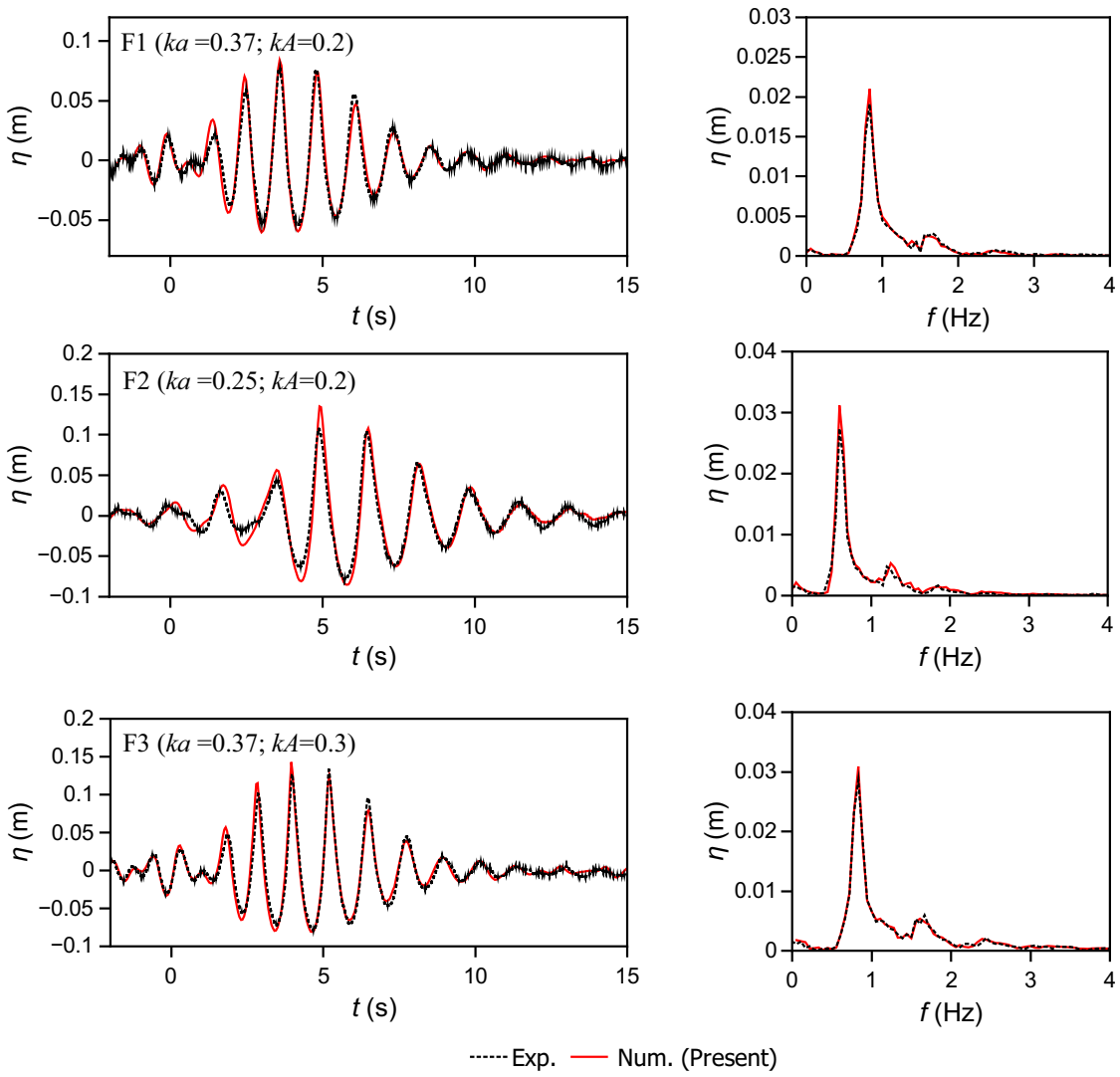


Figure 6-11: Comparisons of focused wave elevations at WP1 (left column) and their corresponding amplitude spectra (right column).

particular, for the case F2, where the slenderness ka is smaller than that in the case F1, the wave elevation and the horizontal wave force tend to be more nonlinear, which is confirmed from the spectra as the ratio of high order harmonics to the first harmonic appears to be larger in the case F2. It seems that, for this case, the present model better captures the energy distributions around the second-order harmonics. Furthermore, for the case F3, where the wave steepness is larger ($kA = 0.3$) and hence the waves are highly nonlinear and close to the breaking point, the present model also exhibits very good agreement with the experiment for both the wave elevation and the wave loading on the cylinder. This demonstrates that, in the present "full particle" PIC model, the use of Lagrangian particles for solving

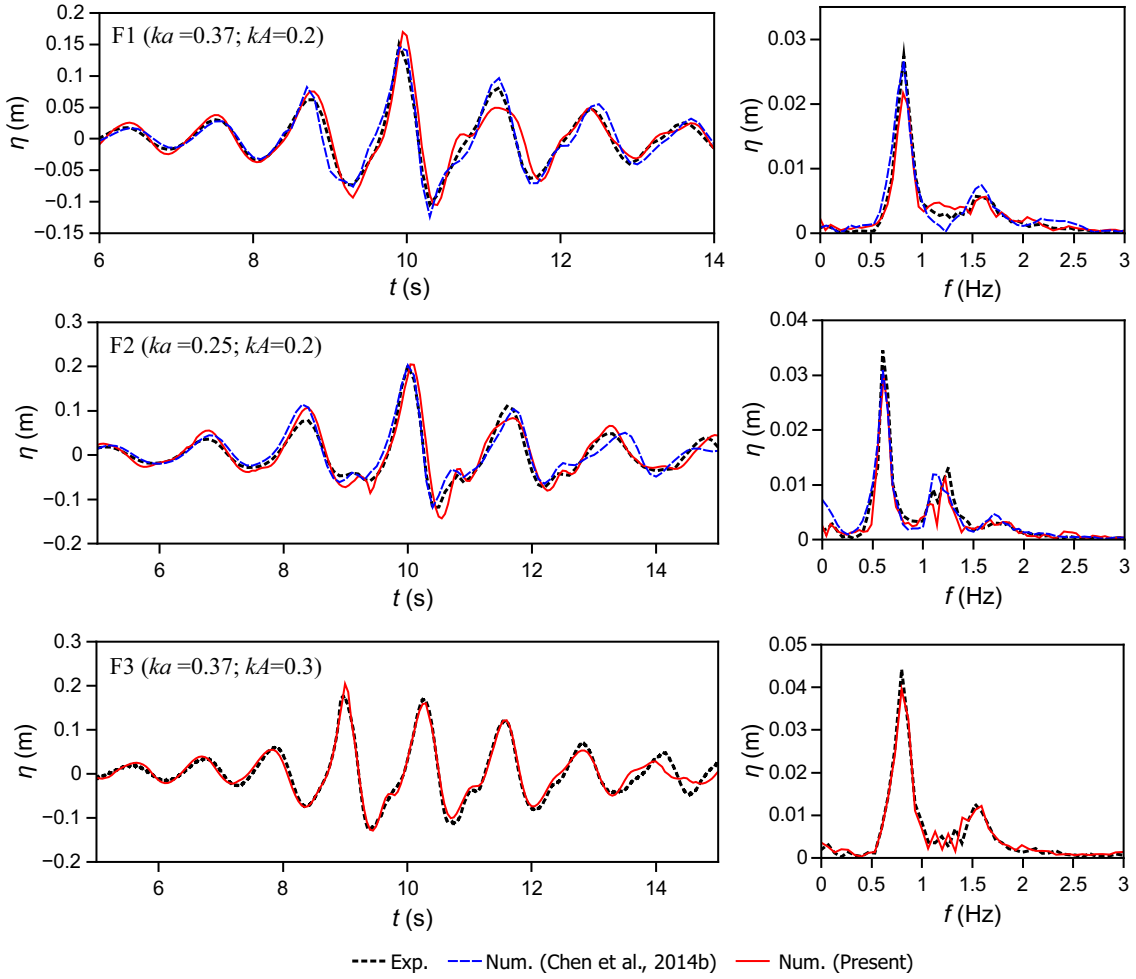


Figure 6-12: Comparisons of focused wave elevations at WP2 (left column) and their corresponding amplitude spectra (right column).

the nonlinear advection term and tracking the free surface is capable of simulating highly nonlinear free-surface waves, as well as their interaction with structures.

It is observed from Figure 6-12 that, particularly for cases F2 and F3 where nonlinearity is strong, small step-shaped waves tend to occur after the focused wave peaks pass the cylinder; for example, around $t = 10.85$ s in the test case F2. This can be explained through snapshots of the present modelling shown in Figure 6-14, from which it can be seen that, at the stagnation point in front of the cylinder, the wave run-up and subsequent strong wash-down of water on the cylinder cause an oscillation of the water surface in front of the cylinder, thus forming the step-shaped waves. These waves are very similar to those predicted by the OpenFOAM® model (c.f. Fig. 14 of Chen et al. (2014b)). It is also observed that, after the primary wave crest hits the cylinder, the wave patterns formed at the lee side of the cylinder are

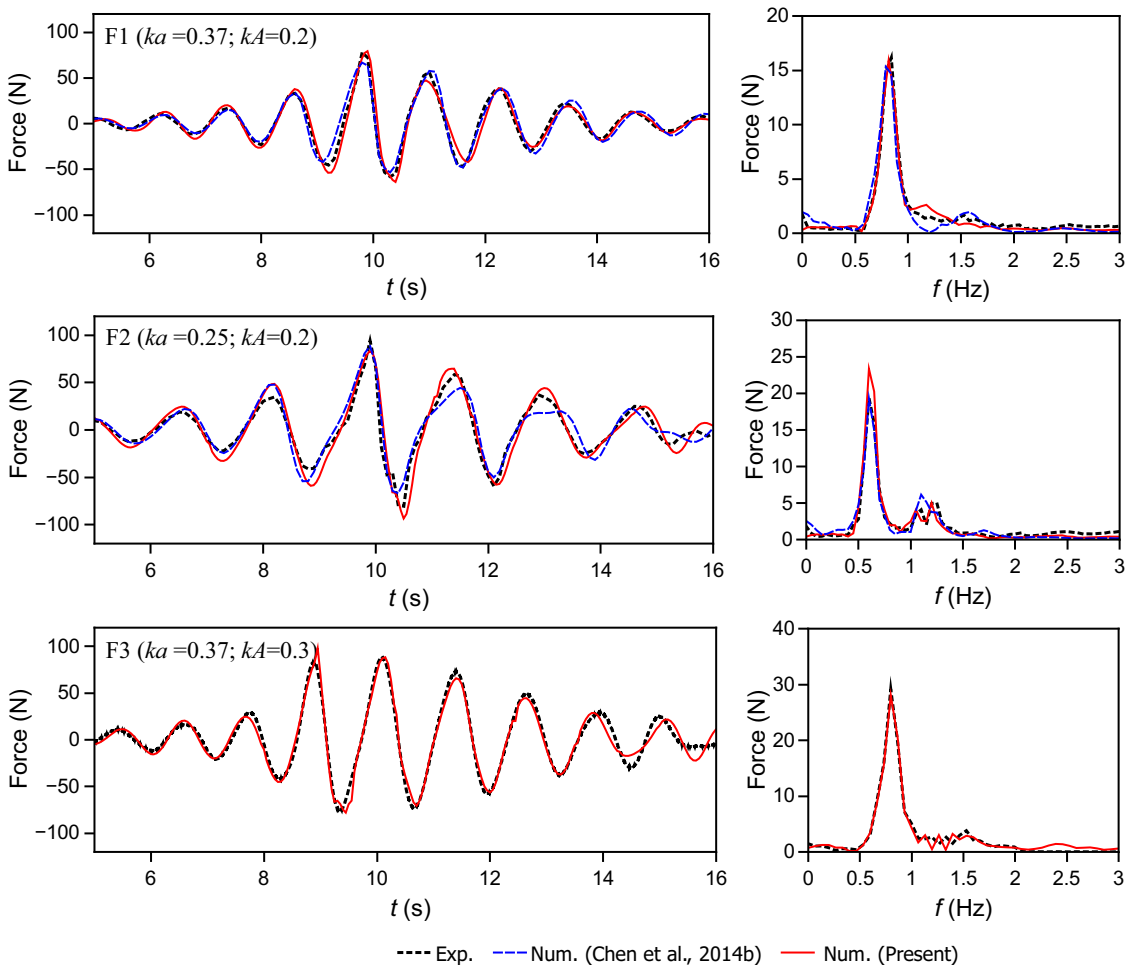


Figure 6-13: Comparisons of the horizontal wave forces on the cylinder (left column) and their corresponding amplitude spectra (right column).

similar to those shown in the regular wave case.

6.3.2 Wave interaction with a group of vertical cylinders

Model setup

This test case demonstrates the capability of the parallel "full particle" PIC model to cope with solitary wave interaction with a group of vertical cylinders. This case was numerically and experimentally studied by Mo and Liu (2009), where both laboratory data and numerical results are available for comparison. In Mo and Liu (2009), the numerical model uses a finite volume scheme coupled with the VOF

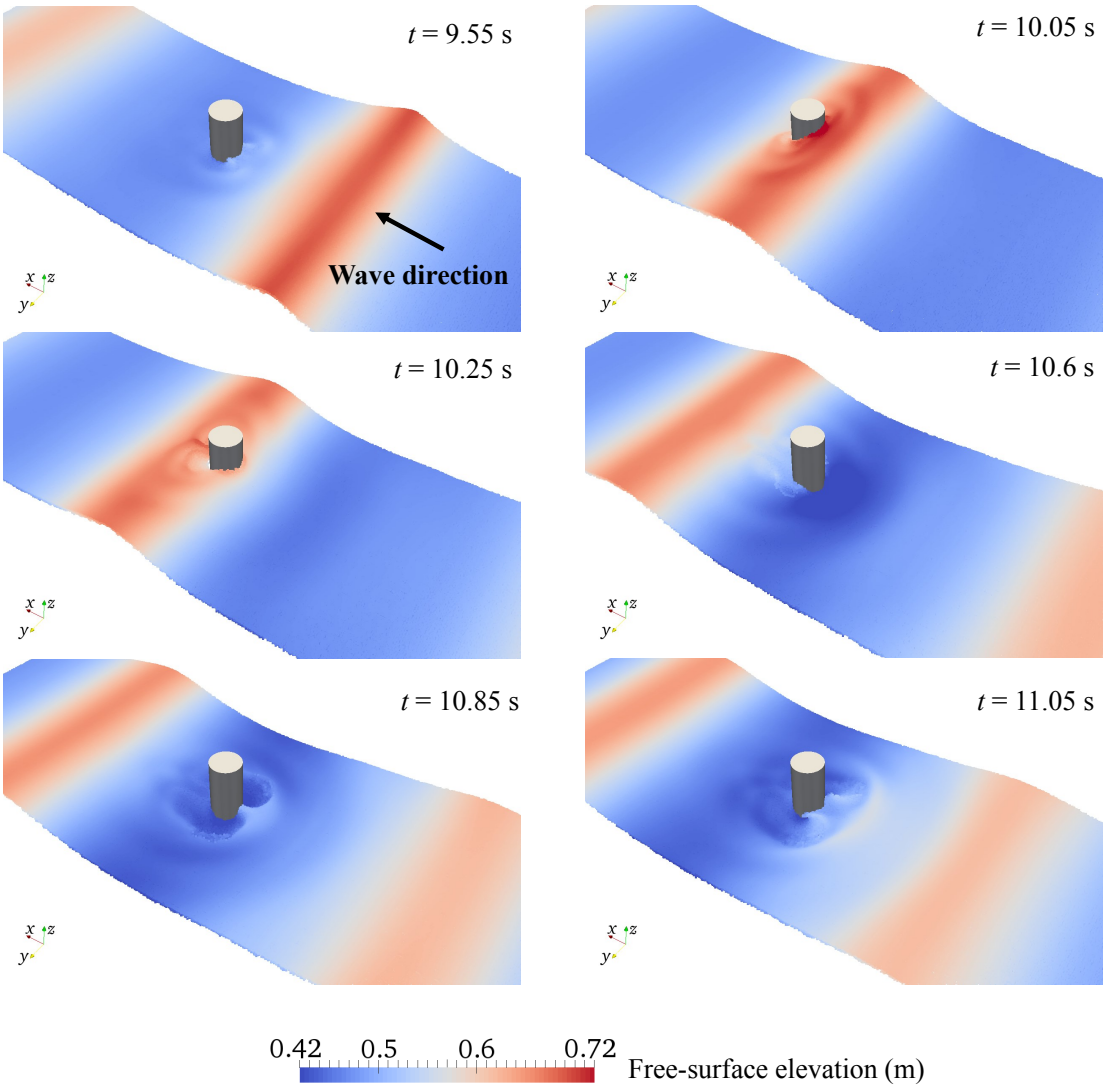


Figure 6-14: Snapshots of the numerical simulation for focused wave (F2) interaction with a single cylinder.

method for free-surface capturing. However, it is noted that their solver is not based on the OpenFOAM® suite. Moreover, in that paper, the Euler's equations are solved without any dissipative mechanism since the investigated solitary waves are non-breaking. Their experiments were conducted in the Tsunami Wave Basin at the O. H. Hinsdale Wave Research Laboratory (WRL) at Oregon State University (OSU).

In the experiment, either one or three identical stainless steel circular cylinders with a diameter, D, of 1.219 m were instrumented and installed in the basin. Figure 6-15

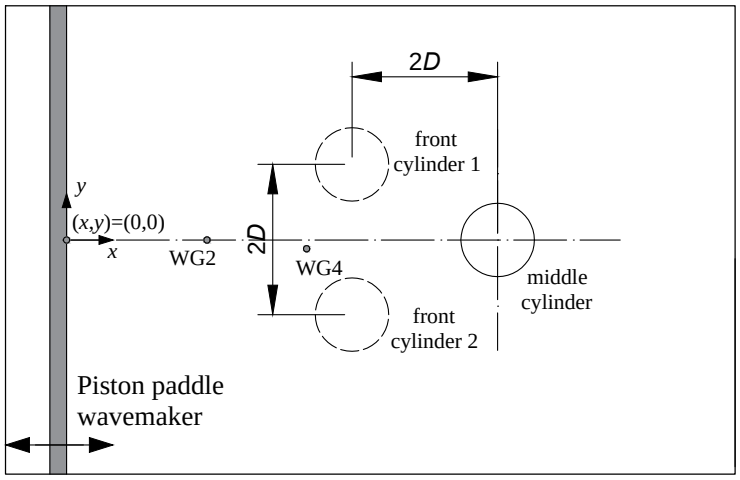


Figure 6-15: Schematic (top view) showing the locations of three identical cylinders. D is the diameter of the cylinder.

shows a sketch of the cylinder locations. The cylinders were symmetrically placed in the basin, with the centre-to-centre distances of $2D$ between the adjacent cylinders in both the streamwise and the spanwise directions. The single cylinder on the downstream side is hereafter referred to as the middle cylinder, and the other two front cylinders are named as the front cylinders. Two test conditions in this section were used: tests of only one cylinder, i.e. the middle cylinder, and all three cylinders. In both the single and the multiple cylinder cases, the water depth, h, was fixed at 0.75 m and the non-breaking solitary wave with a wave height, H, of 0.3 m was used. Wave gauges were used to measure wave elevations around the cylinders, and locations of the two gauges, WG2 and WG4, used in this study are shown in Figure 6-15.

For the numerical modelling, the NWT is 35.0 m long and 12.1 m wide, similar to the dimensions used in Mo and Liu (2009). The origin of the coordinate system is located at the downstream face of the piston-type wavemaker, at the middle of the NWT, as shown in Figure 6-15, with $z = 0$ m indicating the bottom of the tank. The coordinates (in metres) of the centre of the middle cylinder are (17.925, 0), and the wave gauges WG2 and WG4 are located at (11.299, 0) and (14.877, -0.140), respectively. Note that the locations of cylinders and wave gauges in the numerical modelling were shifted to be closer to the wavemaker than those in the experiment to save on CPU cost. The main interests of this test case are the free-surface elevation around the cylinders, wave run-up at the front point of cylinder and wave loading on the cylinder.

The grid size was $\Delta x = \Delta y = \Delta z = 0.03$ m, resulting in, for example, approximately 31.37 million cells with 90.83 million particles (note that the height of the NWT was 2.0 m) for the three cylinders case. The Courant number was 0.5. It took about 10.5 h for $t(g/h)^{1/2} = 47$ of (nondimensional) simulation time with 80 cores at the University of Bath HPCS for the three cylinders case.

Validation of solitary wave generation

The solitary wave was generated using the approaches described in Section 5.5.3, where the initial position of wave crest, ξ_0, was set to -8 m. In the absence of cylinders, Figure 6-16 shows profiles of the solitary wave propagation along the central line of the NWT in x-direction, in comparison with the analytical 9th-order solitary wave solution of Fenton (1972). It can be seen that the numerical solitary wave matches well with the analytical wave, despite that there is a slight change of the wave shape as the solitary wave propagates to further distances, which may be worth a future investigation. Considering that the relative wave height of the solitary wave $H/h = 0.4$ is relatively large, the present results are satisfactory.

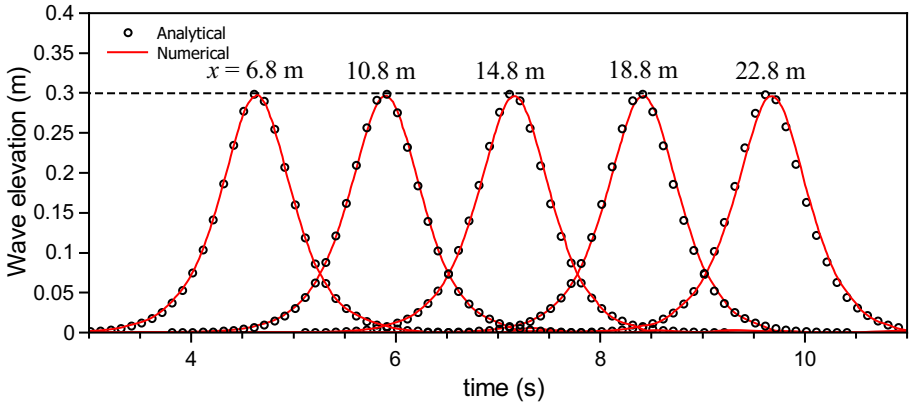


Figure 6-16: Comparison of the solitary wave profiles at different locations for the numerical and the analytical solutions.

Comparison of results for single and three cylinders cases

The solitary wave was reproduced to interact with the cylinders. Figure 6-17 shows snapshots of the current simulations for both a single cylinder and three cylinders cases at the same time instants. In particular, Figure 6-17(B1) and (B3) are plotted at the time instants that maximum run-up occurs at the front cylinders and the

middle cylinder, respectively. For the single cylinder case, it can be seen that as the solitary wave passes the cylinder, it causes wave run-up at the front side, and the water level rise at the lee side due to the symmetric collision between the scattered waves, which then washes frontward down the cylinder. For the three cylinders case, similar processes can be observed for each individual cylinder. Also, it is seen from Figure 6-17(B5) that local wave breaking occurs between the two front cylinders. This is due to an interaction of the symmetrical scattered waves from the front two cylinders as shown in Figure 6-17(B4), combined with the reflected waves from the middle cylinder. This local wave breaking is also noted in Mo (2010). However, it is argued that this relatively small-scale local wave breaking may not be able to significantly influence the whole fluid property fields, and as such turbulence effects due to this local wave breaking may not be predominant in this test case.

The quantitative comparisons for wave elevation, wave run-up at the upstream stagnation point of the middle cylinder and horizontal wave loading on the middle cylinder are shown in Figure 6-18. Results from both the single cylinder and the multiple cylinder cases are presented. In particular, the top panels of Figure 6-18 show the comparisons of the non-dimensional wave elevations, η/H, at WG4 and WG2 for the single and the three cylinders cases, respectively. This is to calibrate the present model for the incident solitary wave generation. It is seen that the numerical results match well with experimental measurements, indicating that the solitary waves generated by the present model are acceptable for this test case.

The middle and the bottom panels of Figure 6-18 show the side-by-side comparisons of the non-dimensional wave run-up, R/H, and the non-dimensional horizontal wave force, $F/\rho g D^3$, at the middle cylinder. For the wave run-up, it is seen that the present results agree very well with the experimental data for both tests. In particular, in the three cylinders case, the present prediction is clearly an improvement on that obtained using the VOF-based solver of Mo and Liu (2009). This hints that the present model correctly simulates the scattered waves from the two front cylinders while they interact with each other and propagate to the front point of the middle cylinder. For the horizontal wave force, it may be seen that generally both numerical models predict similar results. A slight reduction of the peak wave force for the three cylinders case, compared with the single cylinder case, is shown in both the numerical predictions and experimental measurements. This is likely a result of the blockage of the front two cylinders. For the single cylinder case, both numerical models slightly

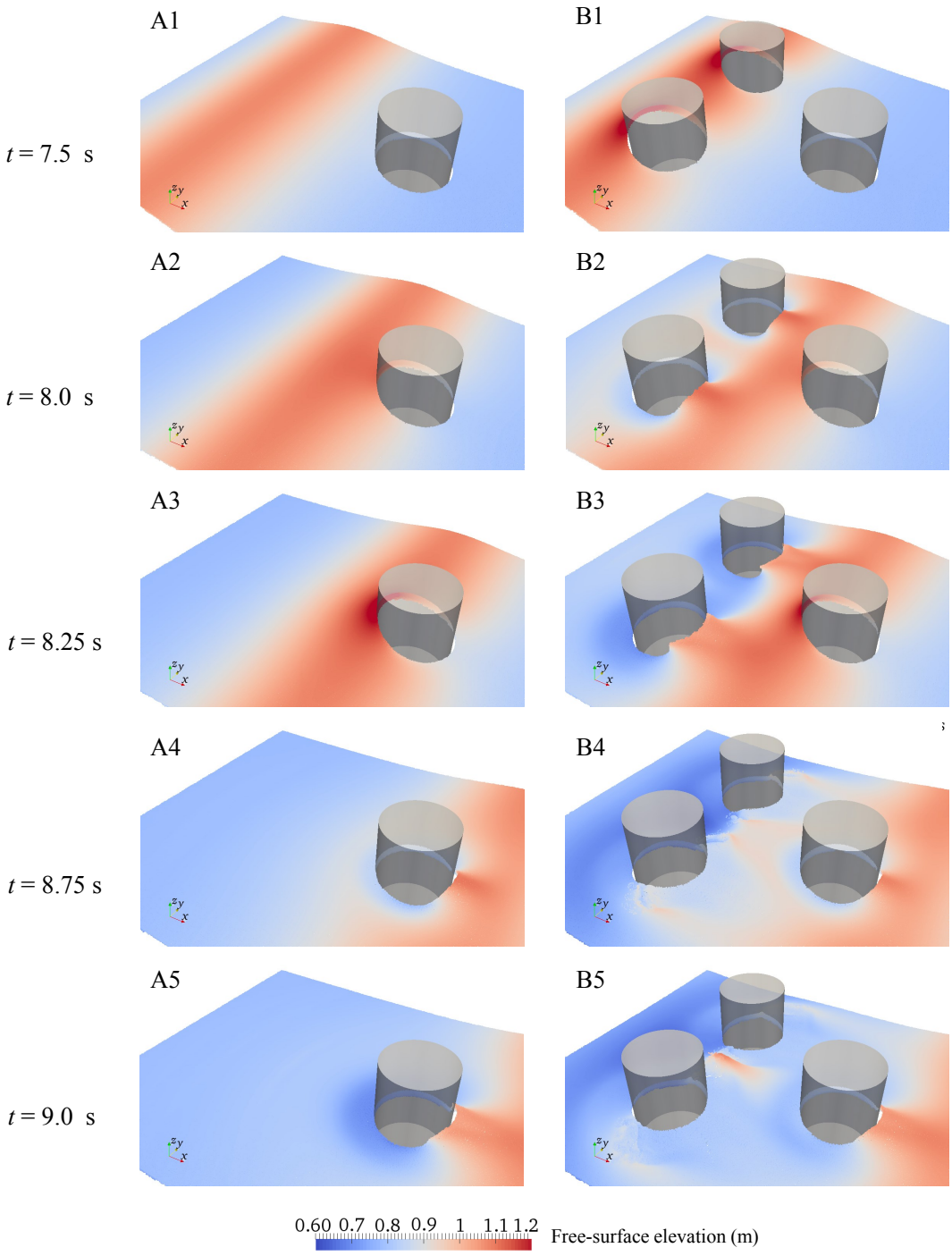


Figure 6-17: Snapshots of the numerical simulations for solitary wave interaction with a single cylinder (A1–A5) and multiple cylinders (B1–B5). The solitary wave propagates from left to right.

underestimate the peak wave force, as well as the trough wave force. In general, the numerical results are in good agreement with the experimental data. It is noted that the experimental data are digitised from Mo and Liu (2009) without selection.

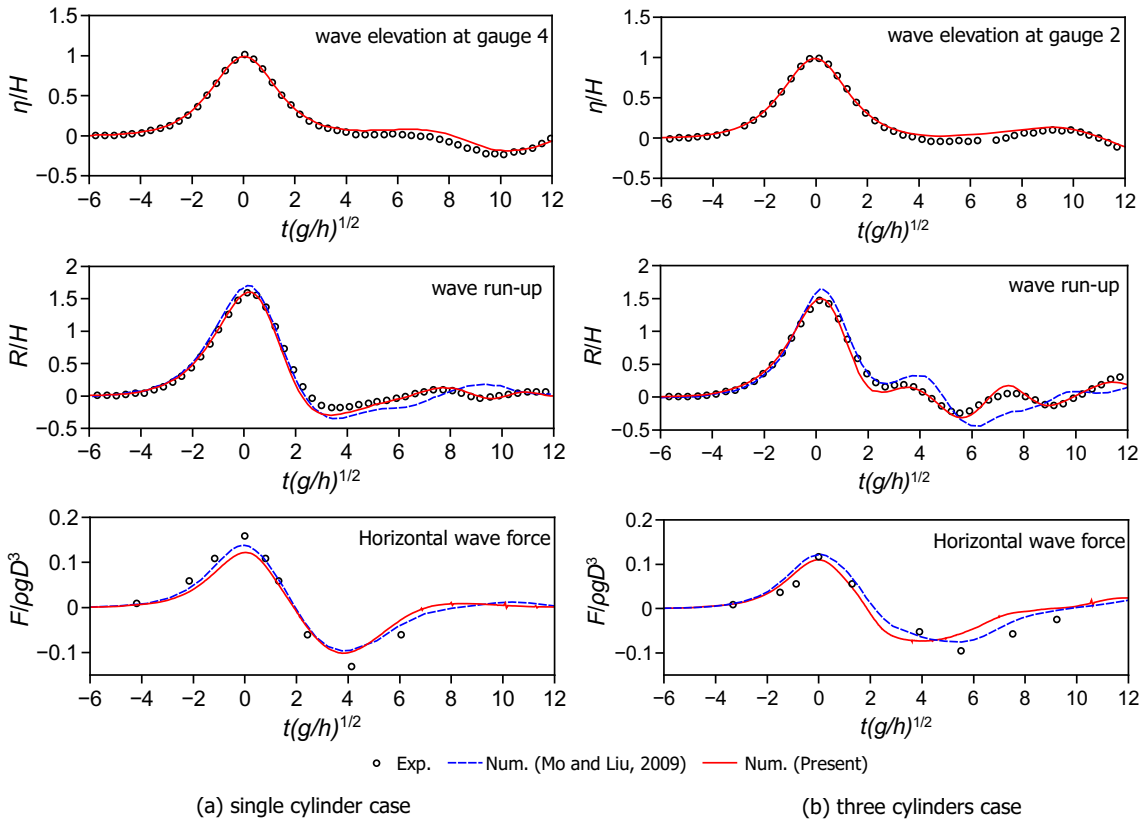


Figure 6-18: Comparisons of free-surface elevation, wave run-up and horizontal wave force at the middle cylinder for both the single and three cylinders cases.

Figure 6-19 compares the wave elevations at WG4 in the three cylinders case for the present model, the VOF-based model of Mo and Liu (2009) and the experiment. This is particularly interesting as the location of WG4 is close to the position at which the local wave breaking occurs as shown in Figure 6-17(B5). It can be seen from Figure 6-19 that the present numerical result matches very well with the experimental measurement. In particular, around the second wave peak that is effected by the local wave breaking the present model appears to perform better. This demonstrates that the use of particles in the present model for tracking the free surface is capable of handling large free-surface deformations such as wave breaking, both straightforwardly and accurately.

Extension to a group of eleven cylinders

To further test the capability of the present solver in simulating more realistic large-scale engineering problems, the case study of a solitary wave interacting with a group of eleven cylinders was conducted. This test case can be regarded as an extension

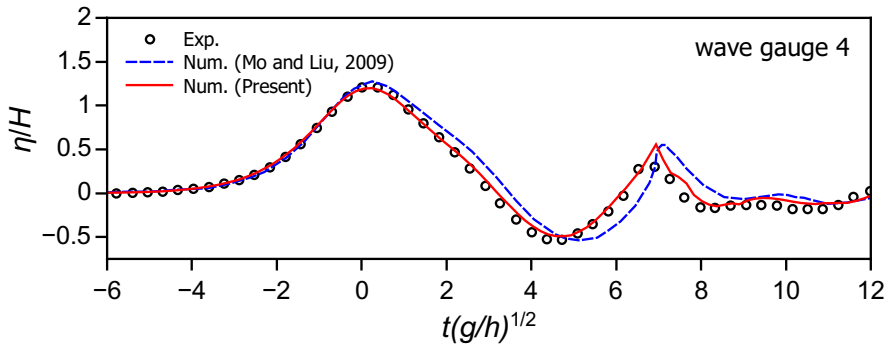


Figure 6-19: Comparison of the free-surface elevation at wave gauge 4 in the three cylinders case.

of the three cylinders case as discussed above. The eleven cylinders were arranged by copying the original set of the three cylinders, with the centre-to-centre distances also being $2D$ between the adjacent cylinders in both the streamwise and spanwise directions. Figure 6-20 depicts a sketch of the setup and the indexes of the cylinders. The same NWT, solitary wave and cylinder size were used. The measured data were the wave run-up and the horizontal wave loading on the cylinders.

Figure 6-21 shows snapshots of the simulated free-surface fields in 2D top view. The solitary wave travels from left to right. From Figure 6-21(a)–(c), it is seen that free-surface evolutions around the three cylinders groups 4-5-6 and 1-2-3 are similar to those of the three cylinders case discussed above. In the meantime, circular scattered waves start to propagate radially away from the cylinders. Also, from Figure 6-21(c)–(e) it is observed that a portion of scattered waves of the cylinder group 1-2-3-4 seems to be trapped, travelling between cylinders 1 and 4. This also occurs in the cylinder group 1-7-8-9 as shown from Figure 6-21(d)–(f). Furthermore, after the solitary wave

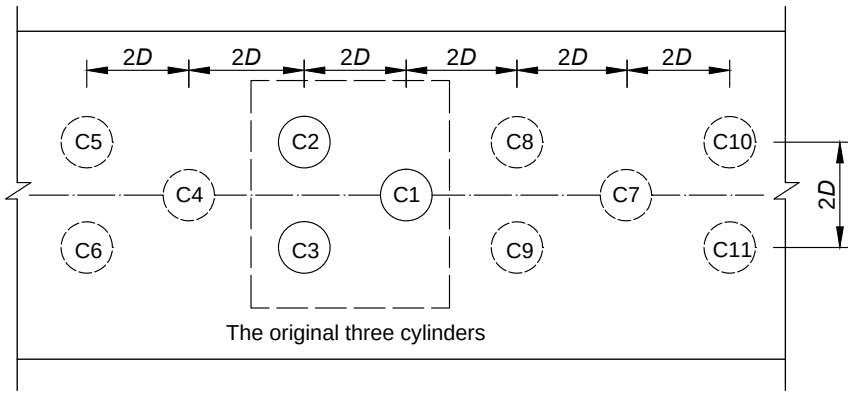


Figure 6-20: Schematic (top view) showing locations of the eleven cylinders.

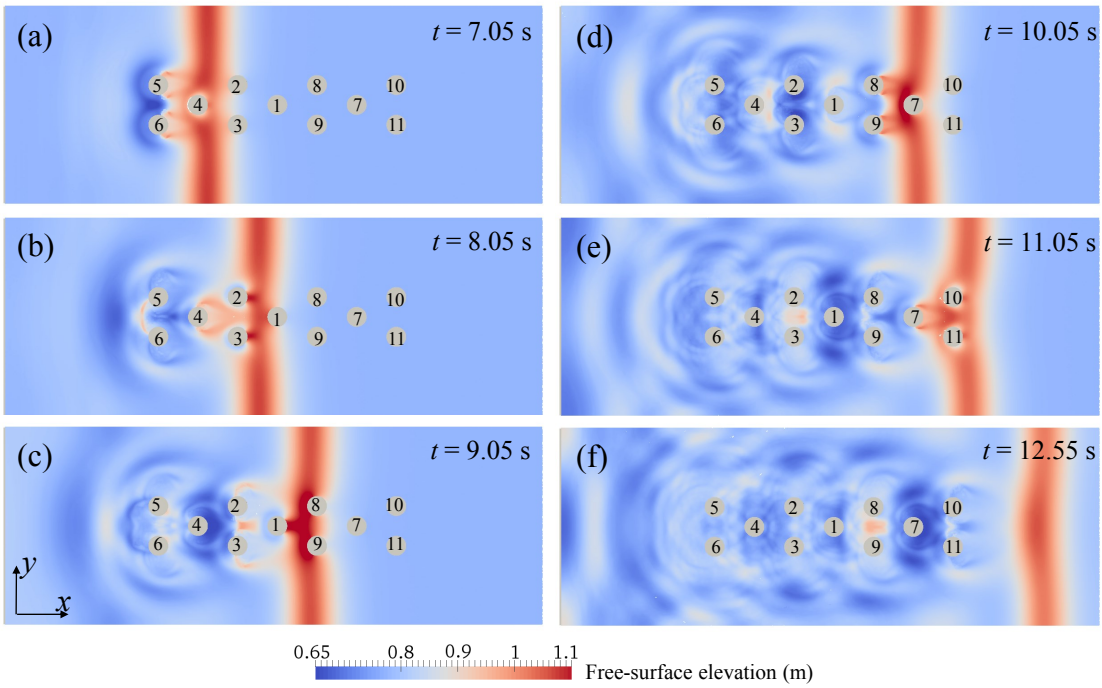


Figure 6-21: Snapshots of the numerical simulation for solitary wave interaction with a group of eleven cylinders.

passes the main body of the cylinder group, a complex wave field has formed as seen from Figure 6-21(f), due to both wave-wave and wave-structure interactions.

Figure 6-22 shows wave run-up at the upstream stagnation points of both middle cylinders 1, 4 and 7, and lateral cylinders 2, 5, 8 and 10. For comparison purposes, the results are all shifted in time, with the time instants of all primary peak values aligned with the non-dimensional time $t(g/h)^{1/2} = 0$. It is seen from Figure 6-22(a) that the middle cylinders encounter very similar wave run-up for this configuration of cylinder locations, although they are blocked by the lateral cylinders. Similarly, as shown in Figure 6-22(b), very similar run-up occurs at the lateral cylinders 2, 8 and 10, except cylinder 5, which is located at the front of the cylinder group. Both the primary and the secondary peaks of the wave run-up at cylinder 5 appear to be the largest. In particular, the secondary peak wave is very sharp and may be close to break, which is caused by the symmetric scattered waves from cylinders 5 and 6 interacting with each other and propagating upstream as seen from Figure 6-21(a) and (b).

Figure 6-23 presents comparisons of the horizontal wave forces on the cylinders. It

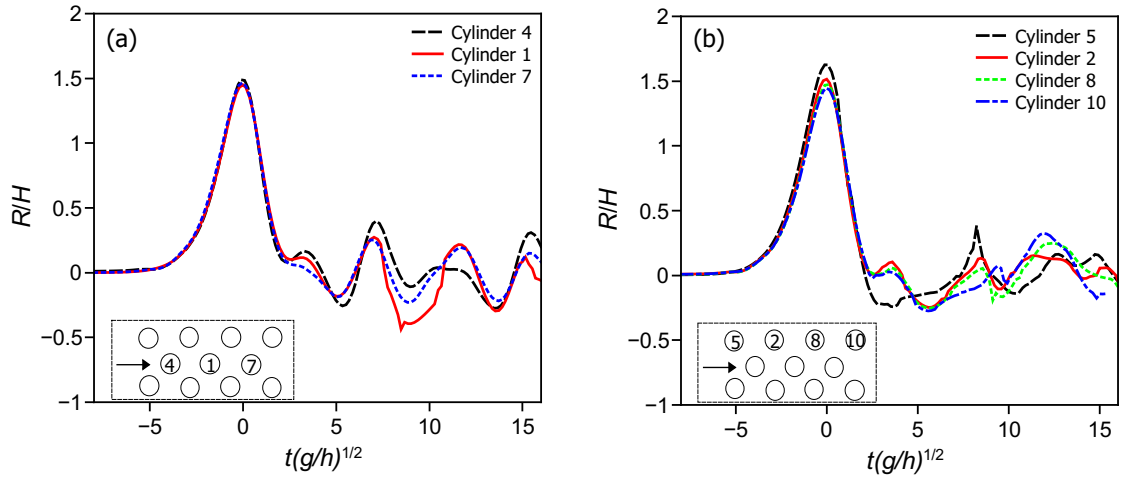


Figure 6-22: Wave run-up at the upstream stagnation points of different cylinders: (a) middle cylinders; (b) lateral cylinders.

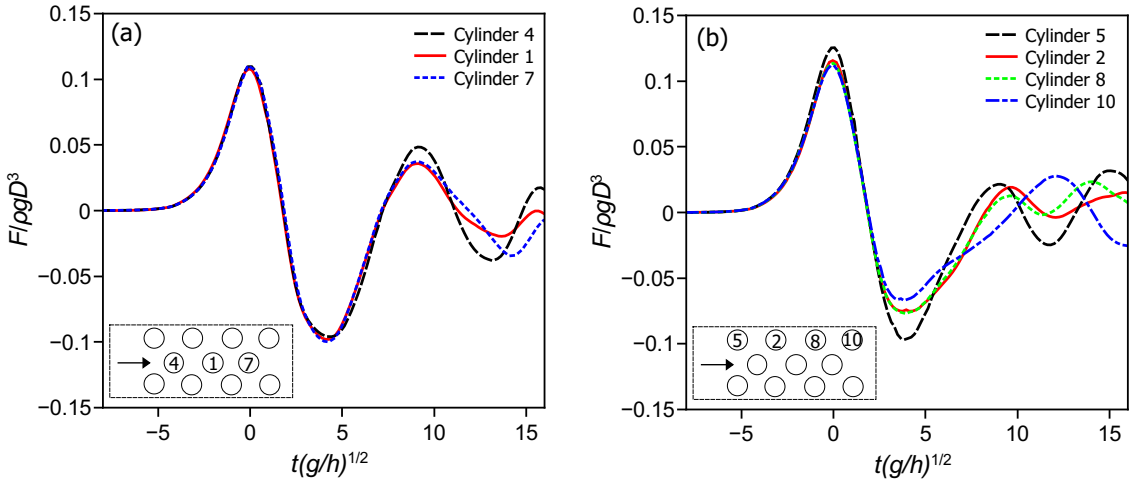


Figure 6-23: Horizontal wave loading on different cylinders: (a) middle cylinders; (b) lateral cylinders.

is seen from Figure 6-23(a) that the horizontal wave forces are nearly identical on all the three middle cylinders until the secondary force peaks. These secondary peak values are approximately 38% of their corresponding primary peak values with respect to the horizontal wave forces. For the forces on the lateral cylinders, as shown in Figure 6-23(b), cylinder 5 again encounters the largest wave forces for both the primary peak and the primary trough. Apart from these, the secondary peak forces on all lateral cylinders are very close and are around 20% of their corresponding primary peak forces.

6.4 Summary

In this chapter, the "full particle" PIC model developed in chapter 3 and chapter 5 was extended to three spatial dimensions and parallelised using the MPI approach. The 3D parallel model was validated against a series of test cases relating to nonlinear wave interaction with vertical cylinders. Numerical tests were performed using regular, focused and solitary waves. Both the single and multiple cylinders cases were simulated. The results of the present model agree well with experimental measurements and those of other state-of-the-art VOF based Eulerian solvers.

For all the validation tests, the present 3D parallel model is shown to be able to reproduce the key physical processes. In particular, the highly nonlinear water waves and small-scale breaking waves which occurred in each of the test cases are well resolved. This appears to be one of the primary benefits arising from the particle nature of the PIC method in tracking the free surface and solving the transport terms of the governing equations. In addition, the MPI parallelisation enables the current PIC model to simulate large-scale 3D problems (i.e. those require a large number of grid cells and particles), with the computational efficiency (in terms of CPU cost) being slightly less than the widely used Eulerian OpenFOAM® model, as demonstrated through a rough comparison in Section 6.3.1. Therefore, it is shown that the present "full particle" PIC model, with new innovations, has great potential to become a high quality CFD tool for use in coastal and offshore engineering applications.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this chapter the major conclusions from the work presented in this thesis are detailed, together with recommendations for future work.

7.1 Conclusions

This thesis presents a novel "full particle" PIC based numerical solver for single-phase free-surface flows, as well as their interaction with coastal and offshore structures. According to the PIC methodology, the model uses both particles and grid to solve the incompressible Newtonian NS equations. The particles carry the material information of the fluid and are employed to solve the nonlinear advection term and track the free surface, while the grid is used to resolve the body force, viscosity term and incompressibility restrictions for computational convenience. The model successfully combines certain advantages of both meshless Lagrangian methods and pure Eulerian methods, and as such is able to simulate complex physical problems such as violent wave impacts on structures, with satisfying flexibility, efficiency (in terms of CPU cost) and accuracy.

The model is developed firstly by establishing a numerical tank for simulating free-surface flows only. Secondly, a tailored DLM method and a Cartesian cut cell based two-way strong coupling algorithm are developed in two spatial dimensions for the modelling of fluid-structure interaction. While the former is primarily employed for modelling fall-pipe rock dumping, the latter is utilised to simulate nonlinear wave-structure interactions typically encountered in the coastal and offshore environment.

Thirdly, a 3D version of the cut cell based fluid-structure interaction model is established and parallelised using the open-source MPI library. Throughout each stage, the model is tested against a variety of benchmark tests.

In chapter 4 it is demonstrated that the "full particle" PIC model, as well as the coupled tailored DLM method, is capable of simulating highly complex problems of rocks entering water and falling down through pipes. The model seems to capture the key elements of this process. The feasibility of this application is mainly due to the fact that the present model incorporates an auxiliary Eulerian grid, which gives access to advanced Eulerian methods, such as the DLM method, for fluid-structure interaction. This is also the case for the development of a Cartesian cut cell based coupling scheme as presented in chapter 5 and chapter 6.

In chapter 3, chapter 5 and chapter 6, a NWT is established within the "full particle" PIC framework in both two and three spatial dimensions, making use of the cut cell based strong coupling algorithm for fluid-structure interaction and wave generation. Different types of water waves, such as regular, focused and solitary waves, and their interaction with either fixed or movable structures of different shapes are studied. The model is shown to have great potential in terms of handling large free-surface deformations, such as wave overtopping and breaking, both straightforwardly and accurately, due to the use of Lagrangian particles. In addition, the model is capable of simulating highly nonlinear water waves and the interaction of such waves with coastal and offshore structures. Particularly, as demonstrated in chapter 5, in two spatial dimensions, the model is able to stably predict the responses of movable structures under either regular or extreme wave conditions.

The present "full particle" PIC model is validated against not only physical experiments but also other state-of-the-art numerical models. In particular, the results are compared with both Eulerian models (e.g. the OpenFOAM® model) and Lagrangian models (e.g. SPH based models). It is found that the present model is able to produce very similar results to those numerical models, and, for some tests, the present model even performs better. For example, in Section 5.6.3 the present model predicts vertical wave forces on a horizontal cylinder better than a state-of-the-art SPH model. As another example, in Section 6.3.2 the present model captures the free-surface elevation effected by a local wave breaking better than a VOF-based finite volume model. In terms of CPU cost, through rough comparisons, it seems

that the present model achieves a computational efficiency that is slightly less than the Eulerian OpenFOAM® model, and is more efficient than the Lagrangian SPH models.

The MPI parallelisation used in chapter 6 mitigates the demanding memory requirement of the present PIC model and thus enables it to run large-scale 3D simulations (e.g. those require a large number of grid cells and particles). In the modelling of more realistic engineering type problems of solitary wave interaction with a group of eleven cylinders as shown in chapter 6, it is illustrated that, with further improvements, the "full particle" PIC model has great potential to become a high quality CFD tool for coastal and offshore engineering applications.

7.2 Recommendations for future work

In general, it is seen that the designed model is convergent, but has a certain amount of numerical dissipation, as the model is, formally, first-order accurate both in time and space. Also, the stability and accuracy of the fluid pressure computation in the small cut-cells still requires improvement. Furthermore, despite being parallelised with the MPI approach, the highly demanding memory requirement is inherent in the PIC methodology, which still limits its development and application when a very fine grid is required. Further work on developing the "full particle" PIC numerical method for fluid-structure interaction is thus recommended as follows:

- Regarding the accuracy of cut cell approach, it is noted, as did in many other researches (e.g. Kirkpatrick et al. (2003) and Qian et al. (2006)), that in cases where a cut cell becomes arbitrarily small, the accuracy and hence stability of cut cell scheme will be affected. Therefore, a consistent cell-merging technique could be developed in the future. Particularly, when the staggered grid is used for solving full NS equations, the cell-merging technique proposed in Kirkpatrick et al. (2003) could be utilised.
- The model in this thesis is designed for single-phase flows only. A two-phase (e.g. water-air) solver could be built such that the model can simulate, for

example, the natural enclosure of air pockets in breaking waves and their influences on wave impacts on structures. A pioneering work in this direction can be found in Boyd and Bridson (2012), where both air and water phases are modelled as incompressible fluids. For modelling air as compressible and water as incompressible, the SPH work of Lind et al. (2015) would provide a suitable reference point.

- Turbulence effects that correspond to physical processes are not included in the present model. The RANS equations, coupled with a transport model for the turbulence properties (e.g. k–epsilon or k–omega) (Wilcox et al., 1998), could be used in future investigations. With a turbulence closure model, the "full particle" PIC model could better resolve the effects of breaking waves.

- The present model was developed on a uniform grid, which, particularly in three spatial dimensions, can lead to an exaggerated number of grid cells and particles and hence unrealistic memory requirements and CPU cost. While in this thesis this problem is mitigated through domain decomposition and parallelisation, an alternative option could be the employment of adaptive grid such as quadtree and Octree grids, coupled with particle merging and splitting. Byproducts of this could be the improvement on the enforcement of the free-surface boundary conditions and a reduction of spikes in pressure signals when simulating violent wave impacts. The work presented in Liang et al. (2007), Hong (2009) and Ando et al. (2012) should be referred to.

- As mentioned above, the present solver is first-order accurate both in time and space. High-order PIC solvers are thus desirable and could be developed for free-surface flows. For improving accuracy in time, high-order projection methods for solving the NS equations such as those proposed in Kim and Moin (1985); Choi and Moin (1994); Yang and Prosperetti (2006) could be exploited, but these methods may require modifications when applied to the PIC framework. For adding spatial accuracy, high-order interpolation schemes for velocity transfer from scattered particles to a uniform grid must be investigated. The method proposed in Liu and Liu (2006) may prove helpful.

REFERENCES

Adalsteinsson, D., Sethian, J. A., 1999. The fast construction of extension velocities in level set methods. Journal of Computational Physics 148 (1), 2–22.

Ando, R., Thurey, N., Tsuruno, R., 2012. Preserving fluid sheets with adaptively sampled anisotropic particles. Visualization and Computer Graphics IEEE Transactions on 18(8), 1202–1214.

Ardekani, A. M., Dabiri, S., Rangel, R. H., 2007. Modified DLM method for finite-volume simulation of particle flow. In: 45th AIAA Aerospace Sciences Meething and Exhibit 8. Reno, Nevada.

Batty, C., Bertails, F., Bridson, R., 2007. A fast variational framework for accurate solid-fluid coupling. ACM Transactions on Graphics (TOG) 26 (3), 100.

Beemsterboer, T. N., 2013. Modelling the immediate penetration of rock particles in soft clay during subsea rock installation, using a flexible fallpipe vessel. Master's thesis, TU Delft, Delft University of Technology.

Birdsall, C. K., Langdon, A. B., 1985. Plasma physics via computer simulation. Taylor & Francis.

Borazjani, I., Ge, L., Sotiropoulos, F., 2008. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. Journal of Computational Physics 227 (16), 7587 – 7620.

Bouscasse, B., Colagrossi, A., Marrone, S., Antuono, M., 2013. Nonlinear water wave interaction with floating bodies in sph. Journal of Fluids and Structures 42, 112–129.

Boyd, L., Bridson, R., 2012. MultiFLIP for energetic two-phase fluid simulation. ACM Transactions on Graphics (TOG) 31 (2), 16.

Brackbill, J., 1988. The ringing instability in particle-in-cell calculations of low-speed flow. Journal of Computational Physics 75 (2), 469 – 492.

Brackbill, J. U., 2005. Particle methods. International Journal for Numerical Methods in Fluids 47 (8-9), 693–705.

Brackbill, J. U., Kothe, D. B., Ruppel, H. M., 1988. FLIP: A low-dissipation, particle-in-cell method for fluid flow. Computer Physics Communications 48 (1), 25–38.

Brackbill, J. U., Ruppel, H. M., 1986. FLIP: A method for adaptively zoned, Particle-In-Cell caluclations of fluid flows in two dimensions. J. Comp. Phys. 65, 314–343.

Bridson, R., 2008. Fluid Simulation for Computer Graphics. A K Peters, Ltd.

Brown, D. L., Cortez, R., Minion, M. L., 2001. Accurate projection methods for the incompressible navier–stokes equations. Journal of computational physics 168 (2), 464–499.

Calderer, A., Kang, S., Sotiropoulos, F., 2014. Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures. Journal of Computational Physics 277, 201 – 227.

Carlson, M., Mucha, P. J., Turk, G., 2004. Rigid fluid: animating the interplay between rigid bodies and fluid. In: ACM Transactions on Graphics (TOG). Vol. 23. ACM, pp. 377–384.

@CEA/DEN, @EDF R&D, @OPEN CASCADE, 2014. SALOME 7.4.0[computer program]. http://www.salome-platform.org/[Accessed 16/05/2015].

Chen, J., Beraun, J., 2000. A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. Computer Methods in Applied Mechanics and Engineering 190 (12), 225 – 239.

Chen, L., Sun, L., Zang, J., Hillis, A. J., 2014a. Numerical simulation of wave-induced roll of a 2-D rectangular barge using OpenFOAM. In: the 29th International Workshop on Water Waves and Floating Bodies. Osaka, Japan.

Chen, L., Zang, J., Hillis, A. J., Morgan, G. C. J., Plummer, A. R., 2014b. Numerical investigation of wave-structure interaction using OpenFOAM. Ocean Engineering 88, 91–109.

Choi, H., Moin, P., 1994. Effects of the computational time step on numerical solutions of turbulent flow. Journal of Computational Physics 113 (1), 1 – 4.

Chorin, A. J., 1967. A numerical method for solving incompressible viscous flow problems. Journal of Computational Physics 2 (1), 12 – 26.

Chorin, A. J., 1968. Numerical solution of the Navier-Stokes equations. Mathematics of computation 22, 745–762.

Chorin, A. J., 1969. On the convergence of discrete approximations to the navier-stokes equations. Mathematics of computation 23 (106), 341–353.

Colagrossi, A., Bouscasse, B., Antuono, M., Marrone, S., 2012. Particle packing algorithm for SPH schemes. Computer Physics Communications 183 (8), 1641 – 1653.

Colagrossi, A., Landrini, M., 2003. Numerical simulalation of interfacial flows by smoothed particle hydrodynamics. J. Comp. Phys. 191, 448–475.

Cundall, P. A., Strack, O. D. L., 1979. A discrete numerical model for granular assemblies. Geotechnique 29 (1), 47–65.

Daly, B. J., Harlow, F. H., Welch, J. E., Sanmann, E. E., 1965. Numerical fluid dynamics using the particle-and-force method. Technical Report LA-3144, Los Alamos Scientific Laboratory, Los Alamos.

Dean, R. G., Dalrymple, R. A., 1991. Water wave mechanics for engineers and scientists. Prentice-Hall.

Dilts, G. A., 1999. Moving-least-squares-particle hydrodynamics I. Consistency and stability. International Journal for Numerical Methods in Engineering 44 (8), 1115–1155.

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

Dimakopoulos, A. S., Cuomo, G., Chandler, I., 2016. Optimized generation and

absorption for three-dimensional numerical wave and current facilities. Journal of Waterway, Port, Coastal, and Ocean Engineering, 06016001.

Dixon, A. G., Salter, S. H., Greated, C. A., 1979. Wave forces on partially submerged cylinders. Journal of the Waterway Port Coastal and Ocean Division 105 (4), 421–438.

Downie, M. J., Bearman, P. W., Graham, J. M. R., 4 1988. Effect of vortex shedding on the coupled roll response of bodies in waves. Journal of Fluid Mechanics 189, 243–261.

Eastwood, J. W., 1986. Particle simulation methods in plasma physics. Computer Physics Communications 43 (1), 89 – 106.

Edwards, E., 2015. Simulating water for computer graphics: particle-in-cell, explicit surfaces, and discontinuous Galerkin. Ph.D. thesis, University of British Columbia.

Edwards, E., Bridson, R., 2012. A high-order accurate particle-in-cell method. Int. J. Numer. Methods Eng. 90 (9), 1073–1088.

Enright, D., Fedkiw, R., Ferziger, J., Mitchell, I., 2002. A hybrid particle level set method for improved interface capturing. Journal of Computational Physics 183, 83–116.

Evans, M. W., Harlow, F. H., 1957. The particle-in-cell method for hydrodynamic calculations. Technical Report LA-2139, Los Alamos Scientific Laboratory, Los Alamos.

Evans, M. W., Harlow, F. H., 1958. Calculation of supersonic flow past an axially symmetric cylinder. Journal of the Aerospace Sciences 25 (4), 269–270.

Faltinsen, O. M., M., L., Greco, M., 2004. Slamming in marine applications. Journal of Engineering Mathematics 48, 187–217.

Fedkiw, R., Stam, J., Jensen, H. W., 2001. Visual simulation of smoke. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, pp. 15–22.

Fedkiw, R. P., 2002. Coupling an eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method. Journal of Computational Physics 175 (1), 200–224.

Fenton, J., 1972. A ninth-order solution for the solitary wave. Journal of Fluid Mechanics 53, 257–271.

Fortes, A. F., Joseph, D. D., Lundgren, T. S., 1987. Nonlinear mechanics of fluidization of beds of spherical particles. Journal of Fluid Mechanics 177, 467–483.

Foster, N., Fedkiw, R., 2001. Practical animation of liquids. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 23–30.

Gao, F., Zang, J., 2014. Numerical simulations of breaking waves at vertical wall. In: The Eleventh ISOPE Pacific/Asia Offshore Mechanics Symposium. International Society of Offshore and Polar Engineers.

Ghasemi, A., Pathak, A., Raessi, M., 2014. Computational simulation of the interactions between moving rigid bodies and incompressible two–fluid flows. Computers & Fluids 94, 1 – 13.

Gibou, F., Fedkiw, R. P., Cheng, L.-T., Kang, M., 2002. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. Journal of

Computational Physics 176 (1), 205 – 227.

Glowinski, R., Pan, T. W., Hesla, T. I., Joseph, D. D., 1999. A distributed Lagrange multiplier/fictitious domain method for particulate flows. Int. J. Multiphase Flow 24, 755–794.

Goda, Y., 1999. A comparative review on the functional forms of directional wave spectrum. Coastal Engineering Journal 41 (01), 1–20.

Goda, Y., Suzuki, T., 1976. Estimation of incident and reflected waves in random wave experiments. Coastal engineering proceedings 1 (15).

Goring, D. G., 1978. Tsunamis–the propagation of long waves onto a shelf. Ph.D. thesis, California Institute of Technology.

Gotoh, H., Ikari, H., Yasuoka, T., 2009. Simulation of armor blocks in front of caisson breakwater by DEM-MPS hybrid model. In: The Nineteenth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers.

Greaves, D. M., 2006. Simulation of viscous water column collapse using adapting hierarchical grids. International Journal for Numerical Methods in Fluids 50 (6), 693–711.

Griebel, M., Dornseifer, T., Neunhoeffer, T., 1998. Numerical Simulation in Fluid Dynamics. Society for Industrial and Applied Mathematics.

Hadžić, I., Hennig, J., Perić, M., Xing-Kaeding, Y., 2005. Computation of flow-induced motion of floating bodies. Applied mathematical modelling 29 (12), 1196–1210.

Harlow, F. H., 1955. A machine calculation method for hydrodynamic problems. Technical Report LAMS-1956, Los Alamos Scientific Laboratory, Los Alamos.

Harlow, F. H., Apr. 1957. Hydrodynamic problems involving large fluid distortions. J. ACM 4 (2), 137–142.

Harlow, F. H., 1964. The Particle–In–Cell computing method for fluid dynamics. In: Alder, B. (Ed.), Methods in Computational Physics. Academic Press, New York, pp. 319–343.

Harlow, F. H., 1988. PIC and its progeny. Computer Physics Communications 48 (1), 1 – 10.

Harlow, F. H., 2004. Fluid dynamics in Group T-3 Los Alamos National Laboratory: (LA-UR-03-3852). Journal of Computational Physics 195 (2), 414 – 433.

Harlow, F. H., Dickman, D. O., Harris, David E., J., Martin, R. E., 1959. Two-Dimensional Hydrodynamic Calculations. Technical Report LA-2301, Los Alamos Scientific Laboratory, Los Alamos.

Harlow, F. H., Welch, J. E., 1965. Numerical calculation of time dependent viscous incompressible flow of fluid with free surface. Physics of Fluids (1958-1988) 8 (12), 2182–2189.

Higuera, P., Lara, J. L., Losada, I. J., 2013a. Realistic wave generation and active wave absorption for Navier–Stokes models: Application to OpenFOAM®. Coastal Engineering 71 (0), 102–118.

Higuera, P., Lara, J. L., Losada, I. J., 2013b. Simulating coastal engineering processes with OpenFOAM®. Coastal Engineering 71 (0), 119–134.

Hirt, C., Shannon, J., 1968. Free-surface stress conditions for incompressible-flow calculations. Journal of Computational Physics 2 (4), 403 – 411.

Hirt, C. W., Amsden, A. A., Cook, J. L., 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics 14 (3), 227–253.

Hirt, C. W., Nichols, B. D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of computational physics 39 (1), 201–225.

Hong, W.-S., 2009. An adaptive sampling approach to incompressible particle-based fluid. Ph.D. thesis, Texas A&M University.

Hu, C., Kashiwagi, M., 2009. Two-dimensional numerical simulation and experiment on strongly nonlinear wave-body interactions. Journal of Marine Science and Technology 14 (2), 200–213.

Hughes, J. P., Graham, D. I., 2010. Comparison of incompressible and weakly-compressible SPH models for free-surface water flows. Jornal of Hydraulic Research 48, 105–117.

Hyman, J. M., 1984. Numerical methods for tracking interfaces. Physica D: Nonlinear Phenomena 12 (1), 396 – 407.

IADC, 2012. Facts about subsea rock installation, an information update from the IADC[Online], Number 3, Available from: https://www.iadc–dredging.com/ul/cms/fck–uploaded/documents/PDF[Accessed 21/01/2015].

Issa, R., Gosman, A., Watkins, A., 1986. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. Journal of Computational Physics 62 (1), 66 – 82.

Issa, R. I., 1986. Solution of the implicitly discretised fluid flow equations by operator-splitting. Journal of computational physics 62, 40–65.

Jacobsen, N. G., Fuhrman, D. R., Fredse, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFoam®. International Journal for Numerical Methods in Fluids 70 (9), 1073–1088.

Jafari, A., Ashgriz, N., 2013. Numerical Techniques for Free Surface Flows: Interface Capturing and Interface Tracking. Springer US, Boston, MA, pp. 1–27.

Jiang, C., Schroeder, C., Selle, A., Teran, J., Stomakhin, A., 2015. The affine particle-in-cell method. ACM Transactions on Graphics (TOG) 34 (4), 51.

Jiang, G.-S., Peng, D.-P., 2000. Weighted ENO schemes for Hamilton–Jacobi equations. SIAM Journal on Scientific computing 21.

Jung, J. H., Yoon, H. S., Chun, H. H., Lee, I., Park, H., 2013. Numerical simulation of wave interacting with a free rolling body. International Journal of Naval Architecture and Ocean Engineering 5 (3), 333 – 347.

Jung, K. H., 2005. Experimental study on rectangular barge in beam sea. Ph.D. thesis, Texas A&M University.

Jung, K. H., Chang, K.-A., Jo, H. J., 2006. Viscous effect on the roll motion of a rectangular structure. Journal of engineering mechanics 132 (2), 190–200.

Kelly, D. M., 2012. Full particle PIC modelling of the surf and swash zones. In: Proc. 33rd Int. Conf. Coast. Eng. A.S.C.E., Santander, pp. 77–92.

Kim, J., Moin, P., 1985. Application of a fractional-step method to incompressible Navier-Stokes equations. Journal of Computational Physics 59, 308–323.

Kim, J. W., Kyoung, J. H., Ertekin, R. C., Bai, K. J., 2006. Finite-element computation of wave-structure interaction between steep stokes waves and vertical cylinders. Journal of waterway, port, coastal, and ocean engineering 132 (5), 337–347.

Kirkpatrick, M. P., Armfield, S. W., Kent, J. H., 2003. A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid. Journal of Computational Physics 184 (1), 1 – 36.

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

Kwak, D., Chang, J. L. C., Shanks, S. P., Chakravarthy, S. R., 1986. A three-dimensional incompressible navier-stokes flow solver using primitive variables. AIAA journal 24 (3), 390–396.

Lai, M.-C., Peskin, C. S., 2000. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. Journal of Computational Physics 160 (2), 705 – 719.

Lara, J. L., Higuera, P., Guanche, R., Losada, I. J., 2013. Wave interaction with piled structures: Application with IH-FOAM. In: ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering. American Society of Mechanical Engineers, pp. V007T08A078–V007T08A078.

Latham, J.-P., Munjiza, A., Garcia, X., Xiang, J., Guises, R., 2008. Three-dimensional particle shape acquisition and use of shape library for DEM and FEM/DEM simulation. Minerals Engineering 21 (11), 797 – 805, discrete Element Methods (DEM) 07Selected papers from Discrete Element Methods (DEM) 07, Brisbane, Australia, August 2007.

Leboeuf, J., Tajima, T., Dawson, J., 1979. A magnetohydrodynamic particle code for fluid simulation of plasmas. Journal of Computational Physics 31 (3), 379 – 408.

Leschka, S., Oumeraci, H., 2014. Solitary waves and bores passing three cylinders-effect of distance and arrangement. Coastal Engineering Proceedings 1 (34), 39.

LeVeque, R. J., 2002. Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics. Cambridge University Press.

Liang, Q., Zang, J., Borthwick, A. G. L., Taylor, P. H., 2007. Shallow flow simulation on dynamically adaptive cut cell quadtree grids. International Journal for Numerical Methods in Fluids 53 (12), 1777–1799.

Lind, S. J., Stansby, P. K., Rogers, B. D., 2016. Fixed and moored bodies in steep and breaking waves using SPH with the Froude–Krylov approximation. Journal of Ocean Engineering and Marine Energy 2 (3), 331–354.

Lind, S. J., Stansby, P. K., Rogers, B. D., Lloyd, P. M., 2015. Numerical predictions of water–air wave slam using incompressible–compressible smoothed particle hydrodynamics. Applied Ocean Research 49, 57–71.

Lind, S. J., Xu, R., Stansby, P. K., Rogers, B. D., 2012. Incompressible smoothed

particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. Journal of Computational Physics 231 (4), 1499 – 1523.

Liu, G. R., Liu, M. B., 2003. Smoothed particle hydrodynamics: a meshfree particle method. World Scientific.

Liu, M., Liu, G., 2006. Restoring particle consistency in smoothed particle hydrodynamics. Applied Numerical Mathematics 56 (1), 19 – 36.

Liu, M., Liu, G., Lam, K., 2003. Constructing smoothing functions in smoothed particle hydrodynamics with applications. Journal of Computational and Applied Mathematics 155 (2), 263 – 284.

Ma, Q. W., 2005a. Meshless local Petrov-Galerkin method for two-dimensional nonlinear water wave problems. Journal of Computational Physics 205 (2), 611 – 625.

Ma, Q. W., 2005b. MLPG method based on rankine source solution for simulating nonlinear water waves. CMES: Computer Modelling in Engineering &Sciences 9 (2), 193–209.

Maljaars, J. M., 2016. A hybrid particle-mesh method for simulating free surface flows. Master's thesis, TU Delft, Delft University of Technology.

Marder, B. M., 1975. GAP–a PIC-type fluid code. Mathematics of Computation 29 (130), 434–446.

Markham, G., Proctor, M. V., 1983. C.E.G.B. report. Tech. Rep. TRPD/L/0063/M82.

Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Le Touze, D., Graziani, G., 2011. δ–SPH model for simulating violent impact flows. Comput. Methods Appl. Mech. Engrg. 200, 1526–1542.

Mayer, S., Garapon, A., Sørensen, L. S., 1998. A fractional step method for unsteady free-surface flow with applications to non-linear wave dynamics. International Journal for Numerical Methods in Fluids 28 (2), 293–315.

McDonough, J. M., 2013. Lectures in Computational Fluid Dynamics of incompressible flow: Mathematics, Algorithms and Implementations[Online], Available from: http://www.engr.uky.edu/ acfd/me691–lctr–nts.pdf [Accessed 07/04/2016].

Meyer, D. W., Jenny, P., 2004. Conservative velocity interpolation for PDF methods. Proc. Appl. Math. Mech. 4, 466467.

Mo, W., 2010. Numerical investigation of solitary wave interaction with group of cylinders. Ph.D. thesis, Cornell University.

Mo, W., Liu, P. L.-F., 2009. Three dimensional numerical simulations for nonbreaking solitary wave interacting with a group of slender vertical cylinders. International Journal of Naval Architecture and Ocean Engineering 1 (1), 20 – 28.

Mohd-Yusof, J., 1997. Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. Annual Research Briefs. NASA Ames Research Center= Stanford University Center of Turbulence Research: Stanford, 317–327.

Monaghan, J. J., 1989. On the problem of penetration in particle methods. Journal of Computational Physics 82 (1), 1 – 15.

Monaghan, J. J., Lattanzio, J. C., 1985. A refined particle method for astrophysical problems. Astronomy and astrophysics 149, 135–143.

Morris, J. P., Fox, P. J., Zhu, Y., 1997. Modeling low reynolds number incompressible flows using SPH. Journal of Computational Physics 136 (1), 214 – 226.

Nakahara, A., Isoda, T., 1997. 1/f α density fluctuation at the slugging transition point of granular flows through a pipe. Physical Review E 55 (4), 4264.

Nematbakhsh, A., Olinger, D. J., Tryggvason, G., 2013. A nonlinear computational model of floating wind turbines. Journal of Fluids Engineering 135 (12), 121103.

Ng, Y. T., Min, C., Gibou, F., 2009. An efficient fluid–solid coupling algorithm for single-phase flows. Journal of Computational Physics 228 (23), 8807–8829.

Nichols, B. D., Hirt, C. W., 1971. Improved free surface boundary conditions for numerical incompressible-flow calculations. Journal of Computational Physics 8 (3), 434 – 448.

Nishiguchi, A., Yabe, T., 1982. Finite-sized fluid particle in a nonuniform moving grid. Journal of Computational Physics 47 (2), 297 – 302.

Nishiguchi, A., Yabe, T., 1983. Second-order fluid particle scheme. Journal of Computational Physics 52 (2), 390–413.

Noh, W. F., 1963. CEL: A time-dependent, two-space-dimensional, coupled eulerian-lagrange code. Tech. rep., Lawrence Radiation Lab., Univ. of California, Livermore.

Noh, W. F., Woodward, P., 1976. SLIC (simple line interface calculation). In: Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Twente University, Enschede. Springer, pp. 330–340.

Oger, G., Doring, M., Alessandrini, B., Ferrant, P., 2006. Two-dimensional SPH simulations of wedge water entries. Journal of Computational Physics 213 (2), 803–822.

Ohyama, T., Kioka, W., Tada, A., 1995. Applicability of numerical models to nonlinear dispersive waves. J. Coastal Eng. 24, 297–313.

Oliveira, T. C. A., Sánchez-Arcilla, A., Gironella, X., 2012. Simulation of wave overtopping of maritime structures in a numerical wave flume. Journal of Applied Mathematics 2012.

Oñate, E., Idelsohn, S. R., Celigueta, M. A., Suárez, B., 2013. The Particle Finite Element Method (PFEM). An effective numerical technique for solving marine, naval and harbour engineering problems. In: Eça, L., Oñate, E., García-Espinosa, J., Kvamsdal, T., Bergan, P. (Eds.), MARINE 2011, IV International Conference on Computational Methods in Marine Engineering: Selected Papers. Dordrecht: Springer Netherlands, pp. 65–81.

Osher, S., Sethian, J. A., 1988. Fronts propagating with curvature-dependent speed: algorithms based on hamilton–jacobi formulations. Journal of computational physics 79, 12–49.

Patankar, N., Singh, P., Joseph, D., Glowinski, R., Pan, T.-W., 2000. A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. Int. J. Multiphase Flow 26, 1509–1524.

Patankar, N. A., 2001. A formulation for fast computations of rigid particulate flows. Center for Turbulence Research Annual Research Briefs 2001, 185–196.

Patankar, S. V., 1980. Numerical heat transfer and fluid flow. CRC Press.

Patankar, S. V., Spalding, D. B., 1972. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. International Journal of Heat and Mass Transfer 15, 1787–1806.

Paulsen, Bo T., Bredmose, H., Bingham, H. B., Jacobsen, N. G., 2014. Forcing of a bottom-mounted circular cylinder by steep regular water waves at finite depth. Journal of Fluid Mechanics 755, 134.

Peskin, C. S., 1972. Flow patterns around heart valves: a numerical method. Journal of computational physics 10 (2), 252–271.

Press, W., Flannery, B., Teukolsky, S., Vetterling, W., 1992. Numerical Recipes: The Art of Scientific Computing (second edition). Cambridge Univ. Press, New York.

Pullen, T., Allsop, N. W. H., Bruce, T., Kortenhaus, A., Schüttrumpf, H., van der Meer, J. W., 2007. EurOtop–Wave Overtopping of Sea Defences and Related Structures: Assessment Manual (Die Kuste version).

Purvis, J. W., Burkhalter, J. E., 1979. Prediction of critical Mach number for store configurations. AIAA J. 17, 1170–1177.

Qian, L., Causon, D. M., Mingham, C. G., Ingram, D. M., 2006. A free-surface capturing method for two fluid flows with moving bodies. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 462 (2065), 21–42.

Quirk, J. J., 1994. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. Computers & Fluids 23 (1), 125–142.

Raafat, T., Hulin, J. P., Herrmann, H. J., 1996. Density waves in dry granular media falling through a vertical pipe. Physical Review E 53 (5), 4345.

Ralston, A., 1962. Runge–Kutta methods with minimum error bound. Mathematics of Computation 16:80, 431–437.

Ramaswamy, B., 1990. Numerical simulation of unsteady viscous free surface flow. Journal of Computational Physics 90 (2), 396–430.

Rider, W. J., Kothe, D. B., 1995. Stretching and tearing interface tracking methods. AIAA paper 95, 1–11.

Rogers, B. D., Dalrymple, R. A., Stansby, P. K., 2010. Simulation of caisson breakwater movement using 2–D SPH. Journal of Hydraulic Research 48, 135–141.

Saad, Y., 2003. Iterative methods for sparse linear systems. Siam.

Scardovelli, R., Zaleski, S., 1999. Direct numerical simulation of free-surface and interfacial flow. Annual Review of Fluid Mechanics 31, 567–603.

Seibold, B., 2008. A compact and fast Matlab code solving the incompressible Navier-Stokes equations on rectangular domains[Online]. Massachusetts Institute of Technology, Available from: http://math.mit.edu/∼gs/cse/codes/mit18086_navierstokes.pdf[Accessed 25/07/2016].

Skillen, A., Lind, S., Stansby, P. K., Rogers, B. D., 2013. Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body–water slam and efficient wave–body interaction. Computer Methods in Applied Mechanics and Engineering 265, 163 – 173.

Smolianski, A., Shipilova, O., Haario, H., 2007. A fast high-resolution algorithm for

linear convection problems: particle transport method. International Journal for Numerical Methods in Engineering 70 (6), 655–684.

Stam, J., 1999. Stable fluids. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., pp. 121–128.

Su, S.-W., Lai, M.-C., Lin, C.-A., 2007. An immersed boundary technique for simulating complex flows with rigid boundary. Computers & fluids 36, 313–324.

Sussman, M., 2003. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. Journal of Computational Physics 187 (1), 110 – 136.

Sussman, M., Puckett, E. G., 2000. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. Journal of Computational Physics 162 (2), 301–337.

Sussman, M., Smereka, P., Osher, S., 1994. A level set approach for computing solutions to incompressible two-phase flow. Journal of Computational Physics 114 (1), 146 – 159.

The OpenFOAM® Foundation, 2014. OpenFOAM–the open source CFD toolbox: User guide (version 2.3.0)[Online], Available from: http://www.openfoam.org/docs/[Accessed 08/10/2014].

Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., Jan, Y.-J., 2001. A front-tracking method for the computations of multiphase flow. Journal of Computational Physics 169, 708 – 759.

Unverdi, S. O., Tryggvason, G., 1992. A front-tracking method for viscous, incompressible, multi-fluid flows. Journal of computational physics 100, 25 – 37.

Van Kan, J., 1986. A second-order accurate pressure-correction scheme for viscous incompressible flow. SIAM Journal on Scientific and Statistical Computing 7, 870–891.

Viecelli, J. A., 1971. A computing method for incompressible flows bounded by moving walls. Journal of Computational Physics 8 (1), 119 – 143.

Violeau, D., Rogers, B. D., 2016. Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future. Journal of Hydraulic Research 54 (1), 1–26.

Wang, H., Agrusta, R., van Hunen, J., 2015. Advantages of a conservative velocity interpolation (CVI) scheme for particle-in-cell methods with application in geodynamic modeling. Geochemistry, Geophysics, Geosystems 16 (6), 2015–2023.

Westphalen, J., Greaves, D. M., Raby, A., Hu, Z. Z., Causon, D. M., Mingham, C. G., Omidvar, P., Stansby, P. K., Rogers, B. D., 2014. Investigation of wave-structure interaction using state of the art cfd techniques. Open Journal of Fluid Dynamics 4, 18–43.

Whitehouse, R. J. S., Harris, J. M., Sutherland, J., Rees, J., 2011. The nature of scour development and scour protection at offshore windfarm foundations. Marine Pollution Bulletin 62 (1), 73 – 88.

Wilcox, D. C., et al., 1998. Turbulence modeling for CFD. Vol. 2. DCW industries La Canada, CA.

Wu, N.-J., Tsay, T.-K., Chen, Y.-Y., 2014. Generation of stable solitary waves by a piston-type wave maker. Wave Motion 51 (2), 240 – 255.

Yang, B., Prosperetti, A., 2006. A second-order boundary-fitted projection method for free-surface flow computations. Journal of Computational Physics 213 (2), 574 – 590.

Youngs, D. L., 1982. Time-dependent multi-material flow with large fluid distortion. Numerical methods for fluid dynamics 24 (2), 273–285.

Zang, J., Taylor, P. H., Morgan, G., Tello, M., Grice, J., Orszaghova, J., 2010. Experimental study of non-linear wave impact on offshore wind turbine flundations. In: Proceedings of the 3rd International Conference on the Application of Physical Modelling to Port and Coastal Protection. Barcelona, Spain.

Zhao, H., 2004. A fast sweeping method for Eikonal equations. Mathematics of Computation 74, 603–627.

Zhao, X., Hu, C., 2012. Numerical and experimental study on a 2-D floating body under extreme wave conditions. Applied Ocean Research 35, 1–13.

Zhao, X., Ye, Z., Fu, Y., Cao, F., 2014. A CIP-based numerical simulation of freak wave impact on a floating body. Ocean Engineering 87, 50–63.

Zhou, J. G., Stansby, P. K., 1999. An arbitrary Lagrangian–Eulerian σ (ALES) model with non–hydrostatic pressure for shallow water flows. Computer Methods in Applied Mechanics and Engineering 178 (12), 199–214.

Zhou, J. T., 2010. Numerical investigation of breaking waves and their interactions with structures using MLPG_R method. Ph.D. thesis, City University London.

Zhu, S., Moule, G., 1996. An efficient numerical calculation of wave loads on an array of vertical cylinders. Applied Mathematical Modelling 20 (1), 26 – 33.

Zhu, Y., Bridson, R., Jul. 2005. Animating sand as a fluid. ACM Trans. Graph. 24 (3), 965–972.