An Improved Incompressible Smoothed Particle Hydrodynamics Method and Its Application in Free-Surface Simulations

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy in the
Faculty of Engineering and Physical Sciences
December 2009

Rui Xu

School of Mechanical, Aerospace and Civil Engineering

Contents

List of Figures

List of Tables

Abstract

Several incompressible SPH (ISPH) methods have been proposed in the past decade. Here only the widely-used projection method for the pressure and velocity coupling is considered. Based on open source code SPHysics [121], the stability and accuracy of three methods which enforce either a divergence-free velocity field, density invariance, or their combination are tested here through the standard Taylor-Green and spin-down vortex problems. It is shown that the divergence-free ISPH method cannot maintain stability in certain situations although it is accurate before instability sets in. The density invariant ISPH method is stable but inaccurate and noisy. The combined ISPH, combining advantages in divergence-free ISPH and density-invariant ISPH, can maintain accuracy and stability but at a high computational cost. A new divergence-free ISPH approach is proposed here which maintains accuracy and stability without increasing computational cost by slightly shifting particles away from streamlines while correcting their hydrodynamic characteristics. This avoids the highly distorted particle spacings which cause instability. Although this algorithm is not strictly conservative, importantly pressure fields are accurate and noise-free up to highest Reynolds numbers tested. The accuracy and spatial convergence rate have been tested for different operators with the Taylor-Green problem. Several

attempts to improve the accuracy have been conducted. The influence of the Taylor-series correction in the new approach, by considering different terms in Taylor expansion, is also studied. It is shown in this thesis that the ISPH method spatially converges in a first-order way.

Through the benchmark test cases of the lid-driven cavity and bluff body, the algorithm and the code have been validated. For lid-driven cavity case, three different Re values, $Re = 100$, 400 and 1000, have been simulated here. However, it has been shown that ISPH method provides a slower convergence rate, compared to the finite volume method (FVM), which is consistent with the first-order spatial convergence rate, while the pressure predictions from ISPH match the converged finite-volume results very well. For the bluff body, cases with two different Reynolds numbers are simulated here. The lift and drag coefficients are calculated from both ISPH and the FVM code Saturne. With $Re = 20$, both methods give almost identical predictions, but differences appear for higher Reynolds condition, $Re = 100$.

The new ISPH approach has also been applied in free-surface flow simulations. A viscous damping method is proposed to overcome the instability on the free surface caused by truncated kernels. The range of maximum Peclet number for the calculation of the damping viscosity is also suggested, which is demonstrated in this thesis by the collapsing water column problem and other test cases. The instabilities are effectively damped by increasing kinematic viscosity for particles on and adjacent to the free surface. The algorithm is validated against an analytical solution for the flow due to an impulsively started plate and (almost) exact solutions for wet bed dam break problems at zero and small times. The method is also validated for progressive regular waves with paddle motion defined by linear theory. The waves

propagate without decay. The highly accurate predictions show that increasing surface viscosity locally by an order of magnitude does not reduce the accuracy by damping the surface instabilities, generated by the inevitable errors associated with truncated kernels. The quantitative validation is believed to be most thorough for this purpose. Good agreements are obtained for the cases studied.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

4. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://www.campus.manchester.ac.uk/medialibrary/policies/intellectual-property.pdf), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's policy on presentation of Theses.

Acknowledgments

I owe my deepest gratitude to those people who helped me to complete this thesis. My supervisor Professor Peter Stansby (The University of Manchester) gave me encouragement, guidance and support from the initial to the final level enabled me to overcome many difficulties during three-year research. Professor Dominique Laurence (The University of Manchester) inspired me with lots of creative ideas. Dr Charles Moulinec (CCLRC Daresbury Laboratory) did proof reading of this thesis and provided me enormous help on work during PhD years. Dr Juan Uribe (The University of Manchester) supported me on various computing problems. Thanks to the MACE Scholarship from the School of Mechanical, Aerospace and Civil Engineering, University of Manchester. With this financial support, I can finish my PhD research.

I am grateful to Mr Stefano Rolfo and Mr Flavien Billard who shared the house with me during the first year in Manchester and cheered me up with Italian and French jokes. I wish to thank my best friends the Yuan family, the Moulinec family and Mr Nicolas Chini. They helped me get through the difficult times, and provided all the emotional support, comaraderie, entertainment, and caring. I cannot forget the football time with lots of friends, Ali, Nicolas, Eric etc., and the drink sessions with Sarah, Maurice, Murat,

Armando and other friends. Also thanks to Pourya, Roberto and Penny who share the same office with me, give me daily help.

Lastly, and most importantly, I wish to thank my parents, Congzhi Xu and Jiyun Du. They gave life to me, raised me, support me, teach me, and love me. I am also grateful to my wife Haixia Xu. Your love, patience, understanding and encouragement have upheld me.

Chapter 1

Introduction and Thesis Objectives

1.1 Background

FLuid mechanics has two philosophies for the description of fluid flows: Lagrangian and Eulerian. In Lagrangian concept, the observation point moves with the fluid element, with the observer moving with a velocity identical to the fluid element, while in the Eulerian concept, the observer keeps a fixed position without moving, with all the flow quantities as functions of positions and time. The corresponding Computational Fluid Dynamics (CFD) implementations are also classified as the Lagrangian method and Eulerian method. The latter has been well studied for more than fifty years and is widely applied in many aspects of flow simulations. The Eulerian-based approaches are the finite volume method (FVM), the finite difference method (FDM) and the finite element method (FEM). Several commercial

software packages, such as STAR-CD, FLUENT, ABAQUS, for instance, are providing solutions for engineering and hydrodynamic problems. The Lagrangian approach has not been as widely used as the Eulerian one. However, it offers a different possibility over the Eulerian approach. In practical applications, Eulerian based models are not suitable for large deformation of fluids or structures, such as violent free-surface flows and the explosion of solid blocks. Moreover, Lagrangian methods can better prescribe the physical process due to the fundamental similarity with the molecular dynamics [42, 43].

Several Lagrangian methods, such as the vortex method [16], the Smoothed Particle Hydrodynamics (SPH) method [67, 33], the Finite Point or Finite Pointset Method (FPM) [83, 106, 107, 109], the Radial Basis Function Method (RBFM) [50, 51, 79], the Finite Volume Particle Method (FVPM) [39, 80], have been introduced during the last decades. The incompressible SPH (ISPH) method as one of the SPH topics has been heavily studied due to its accurate and efficient numerical performance. The aims of this thesis are to investigate the existing ISPH methods, improving the accuracy and efficiency of these methods and applying the ISPH method for accurate free-surface simulations. The vortex method can date back to 1930s, when Rosenhead [97] calculated Kelvin-Helmholtz instabilities by hand. And the modern developments have been promoted by Chorin [13], Leonard [64] and Rehbach [92]. As an extensively and deeply studied method, the discussion and investigation to the vortex method will not be deployed in this thesis. More details about this method can be found in textbooks [16, 69]. Here, a brief introduction to FVPM, FPM and SPH will be presented. The reasons studying Incompressible SPH (ISPH) method is explained in this chapter.

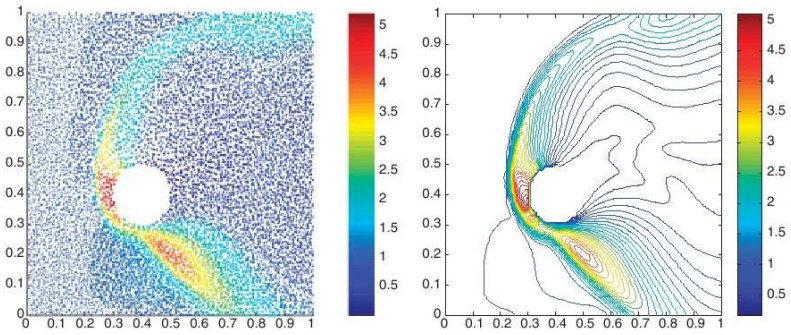
1.2 Finite Volume Particle Method (FVPM)

As a similar particle method for incompressible flow simulations, the Finite Volume Particle Method (FVPM) has been under study almost simultaneously in other different universities and institutions as ISPH method. Good performace has been shown through several academic research cases. Therefore, the author consider it is necessary to give a brief introduction about this method.

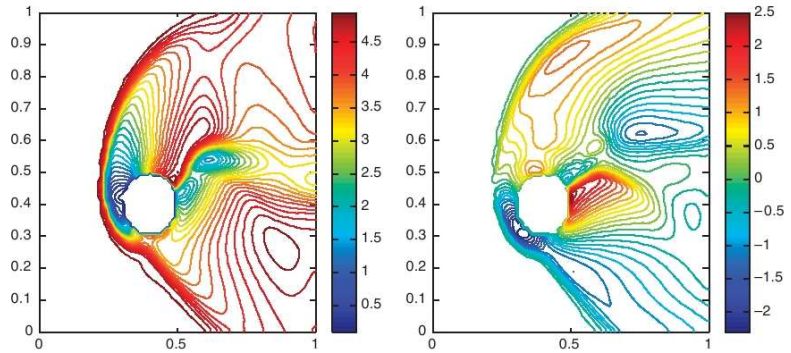
FVPM was first introduced by Hietel et al. [39]. Recent developments have been performed by Keck and Hietel [52, 53], Nestor et. al [80] and Teleaga [104]. In FVPMs, the fluid is represented by a set of particles, which in turn are associated with normalised, overlapping, compactly supported kernel functions. A test function is defined related to the particle volume. The particles are viewed as discrete volumes to which the integral form of the governing equations applies. The interaction between particles is calculated from the flux rate between neighbouring particles. The conservation law is ensured. In contrast to the standard SPH method, the FVPM is conservative regardless of the variation in the particle smoothing lengths. However, the relative advantages of SPH and FVPM under complicated situations for both internal and free-surface flows are not clear. Full comparison between the SPH method and FVPM is needed in future work.

In 2000, Schick [98] applied anisotropic kernel functions and variable kernel supports in the FVPM. Keck and Hietel [53] simulated an inviscid vortex advection problem by applying the projection method [12] in FVPM. In 2008, Teleaga and Struckmeier [105] modelled a compressible inviscid flow around an oscillating circle, as presented in Fig. 1.1. In 2008, Nestor et al. [80] pro-

posed a particle motion correction to prevent the development of poor particle distributions in Lagrangian movement mode, which is also implemented in this thesis with some modifications. Nestor and Quinlan [81] simulated the flow around a circular cylinder with two different local refined particle distributions, as shown in Fig. 1.2. Promising potential for future complicated problems is presented in [81]. The research about free-surface simulations with FVPM however has not been reported, while this has been presented in [100, 101, 54] with ISPH methods. Herewith, the ISPH method is studied here for the purpose of free-surface predictions.

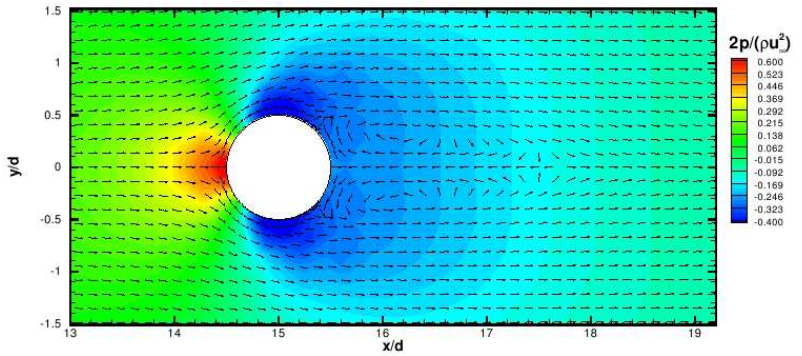


(a) Quasi-random distributed particles and their corresponding density (left) and isolines of the density (right)

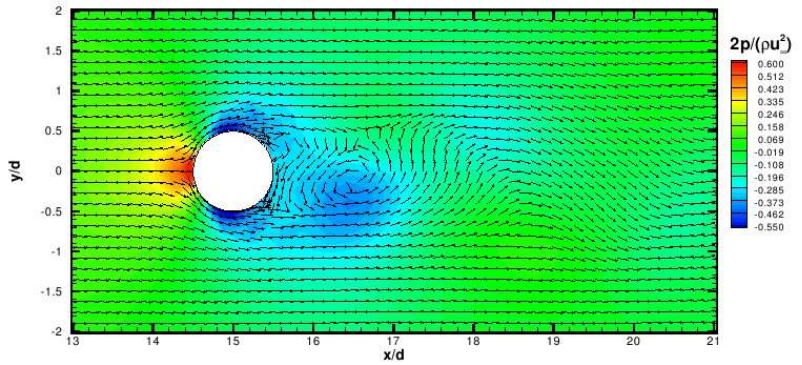


(b) Isolines of the u- (left) and v-velocity components (right) in the same case as in (a)

Figure 1.1: Density and velocity graphs around an oscillating circle [105].



(a) Re=40



(b) Re=100

Figure 1.2: Pressure contours and velocity vectors around a circular cylinder in the fluid field [81].

1.3 Finite Pointset Method (FPM)

In the Finite Pointset Method (FPM) [83, 106, 107, 109], all the points in the computing domain are only interpolation bases, without any relation to the physical properties of the fluid, such as density and volume. The first or second derivative of a hydrodynamic variable is obtained by minimizing the quadratic form of the error function which is built up from the Taylor expansion of the computing point value from the neighbouring points. The details of this method can be found in [83, 106, 107, 109].

FPM was only recently proposed in [83], but the promising potential has

been shown through its high accuracy prediction with disordered point distributions. The application of this method is mainly for fluid simulations. In [83], Oñate et al. simulated inviscid compressible flow around a NACA0012 profile, shown in Fig. 1.3. In [107], the Rayleigh–Taylor instability is predicted with the consideration of the multiphase effect and surface tension, presented in Fig. 1.4. By applying the Incremental Pressure Projection Algorithm (IPPA) [7], Vacondio and Mignosa [109] successfully simulated free-surface flows with FPM, as shown in Fig. 1.5.

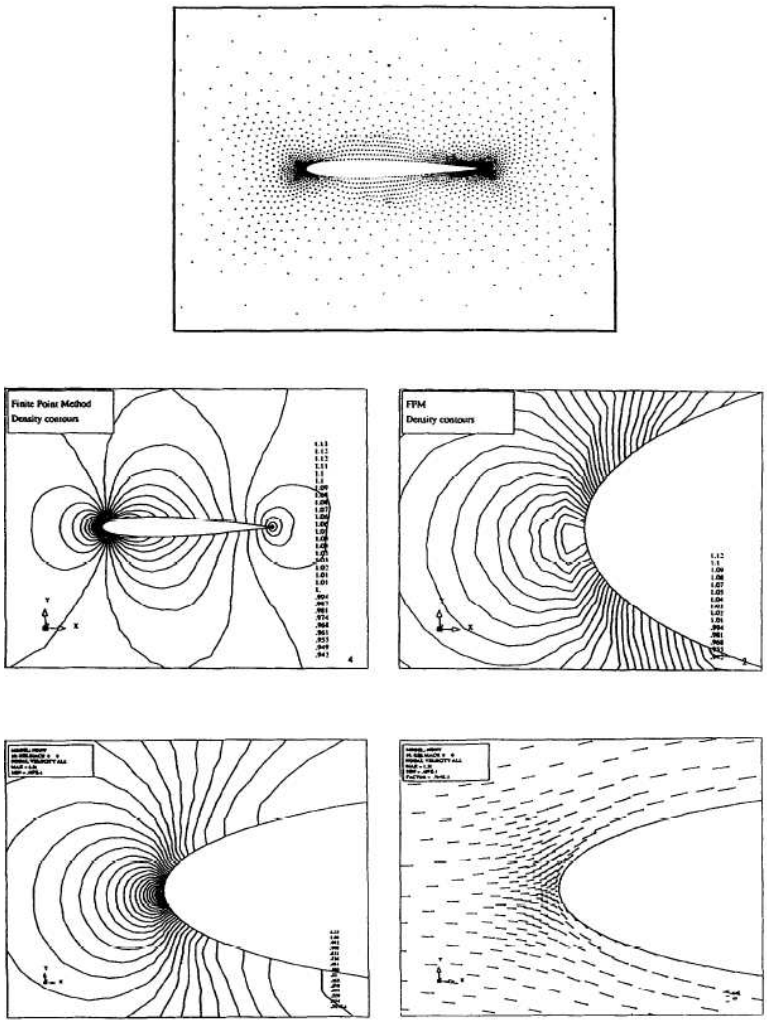


Figure 1.3: Simulation of the inviscid compressible flow around a NACA0012 profile with FPM [83].

The advantage of FPM is that it is a meshless method; any boundary conditions can be enforced analytically by the Taylor expansion without the help of extra particles outside of the simulation domain, and the accuracy of this method can be theoretically increased with extra computing cost by considering more terms in the Taylor expansion for each particle. The stability of the solution is preserved by adding or removing particles when the particle distribution is highly distorted. However, in this way conservation properties are violated.

The biggest disadvantage of FPM is that it is computationally more expensive than the Smoothed Particle Hydrodynamics (SPH) method. In the SPH method, the first or second derivatives can be simply interpolated from the hydrodynamic variable values of the neighbouring particles as presented in §2.1 in Chapter 2, while in FPM the values of the first or second derivatives are obtained from the solution of a linear system of equations based on least-square approximation for each particle, where a matrix with 6×6 entries for 2D and 9×9 entries for 3D are inverted to ensure a second-order accuracy. Also a local iteration approach based on least-square approximation is used to force boundary conditions and satisfy the Pressure Poisson Equation (PPE) in the pressure-velocity coupling. The linear system is repeatedly solved for each particle within the iteration until a converged pressure field is obtained. All those computing procedures make FPM very time consuming.

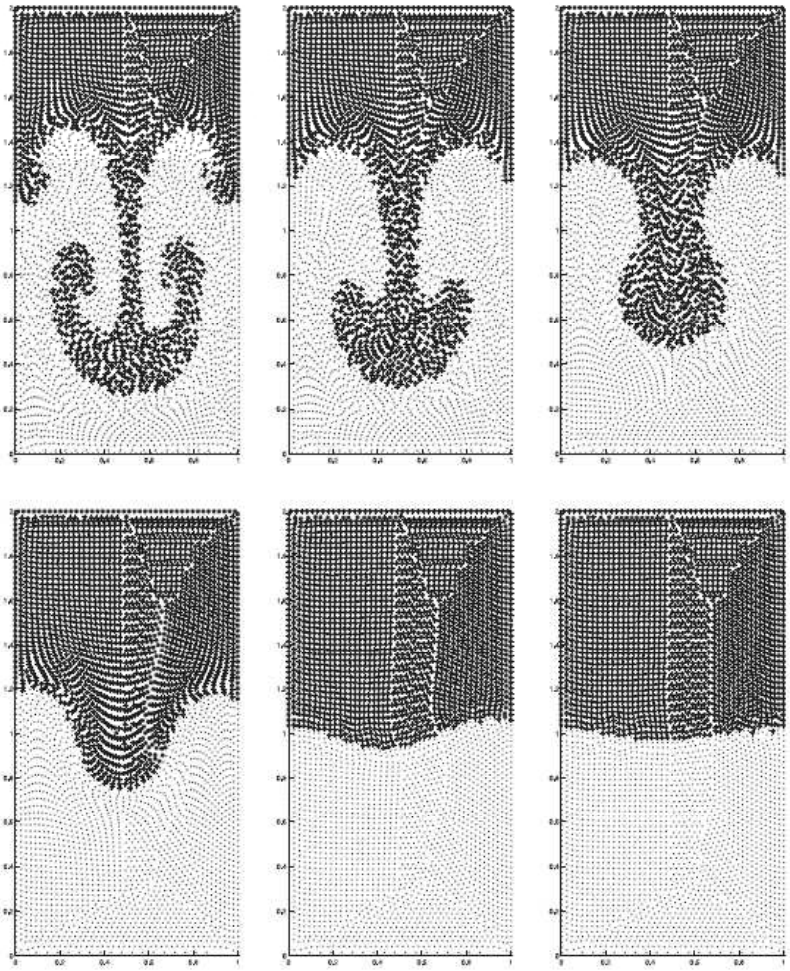


Figure 1.4: Rayleigh–Taylor instability prediction by Tiwari with consideration of the surface tension and multiphase effect [107].

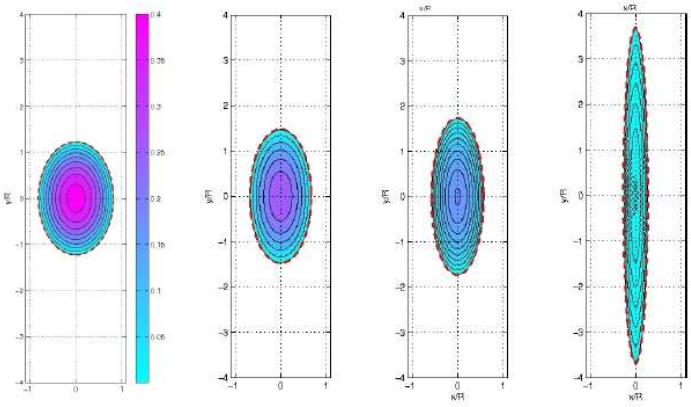


Figure 1.5: Evolution of an initially circular fluid patch: FPM solution of the pressure field at times = 0.2, 0.4, 0.6, 2.0 s [109].

1.4 Smoothed Particle Hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian method, where the fluid medium is discretized by interaction between particles rather than mesh cells. It was first introduced by Lucy [67], and Gingold and Monaghan [71]. The use of SPH has since widely expanded in fluid dynamics and solid mechanics [76] although its original applications were in astrophysics. The basic concept of SPH is that continuous media are represented by discrete particles with volume, density and mass. The particles have a kernel function to define their range of interaction, and the hydrodynamic variable fields are approximated by integral interpolations. Meshes are not needed in the simulation, which is a major advantage of SPH over Eulerian methods for complex geometries.

1.4.1 SPH applications in astrophysics

The original purpose of the SPH method was to perform astrophysical simulations. In astrophysics, non-linear interaction between cosmic objects and non-linear coupling between different interactions, such as electromagnetic, gravitational, as well as the disordered distribution of cosmic objects make it difficult to use traditional mesh-based methods to predict cosmic phenomena. Due to the mesh-free feature, the SPH method can predict the complicated astrophysical process such as fragmentation, supernovae explosion. Fig. 1.6 shows the evolution of binary system in a star formation process [45]. SPH was also used for the star collisions and galaxy formulation predictions [84].

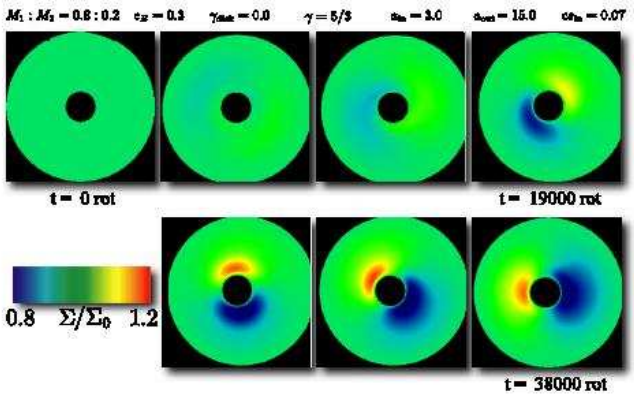


Figure 1.6: The secular evolution of the circumbinary disk [45].

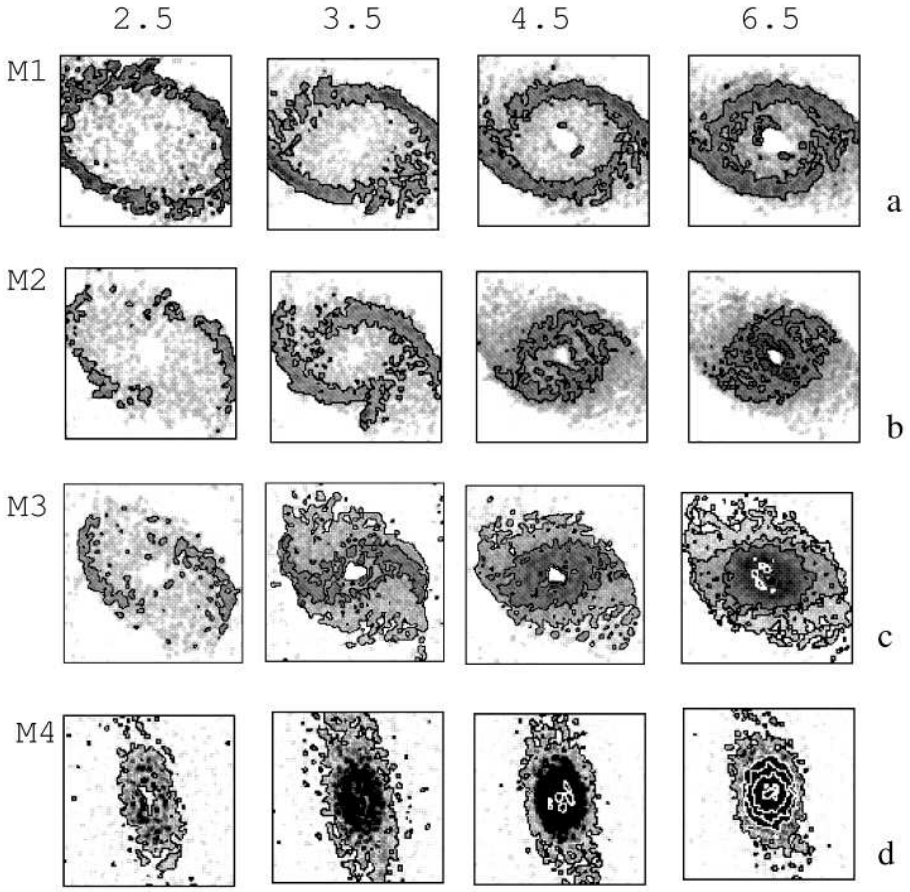


Figure 1.7: Galaxy formulation predicted by the SPH method [84].

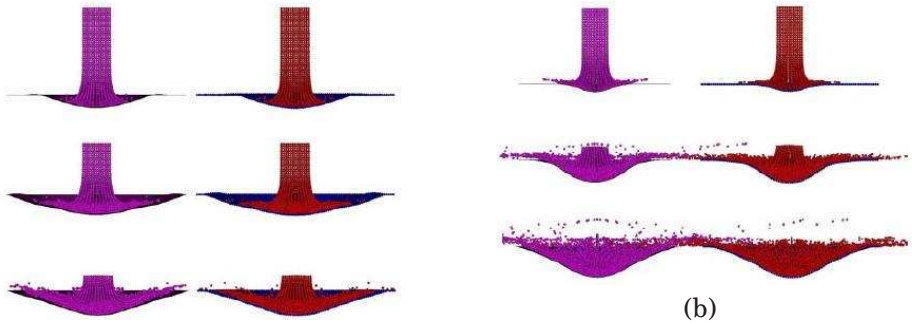


Figure 1.8: Impact of a water column (SPH fluid) on a thin plate: FEM shell (left), SPH shell (right). (a) elastic plate; (b) elastoplastic plate [87].

1.4.2 SPH applications in solid mechanics

Another big field of SPH application resides in solid mechanics. Traditional mesh-based methods, such as Finite Element method (FEM), can perform very well in many situations. However, for high distortion and large deformation, the mesh-based methods cannot provide accurate predictions. With the SPH method, Gray et al. [36] successfully investigated the linear and non-linear oscillation of a plate. Another very interesting work was presented by Potapov et al. [87]. In their work, the fluid impact on a thin elastic/elastoplastic plate was simulated, and the coupling between SPH fluid particles and finite element points was achieved, shown in Fig. 1.8. This work also demonstrated that by only using the SPH method, the fluid-body interaction can be predicted accurately without the help of FEM for the structure change. In 2008, Das and Cleary [21] used the SPH method to successfully simulate the 3D fracture and fragmentation in a thin plate, as shown in Fig. 1.9. The advantage of SPH, handling the large fragmentation without mesh distortion problems, was fully presented in Das and Cleary's work.

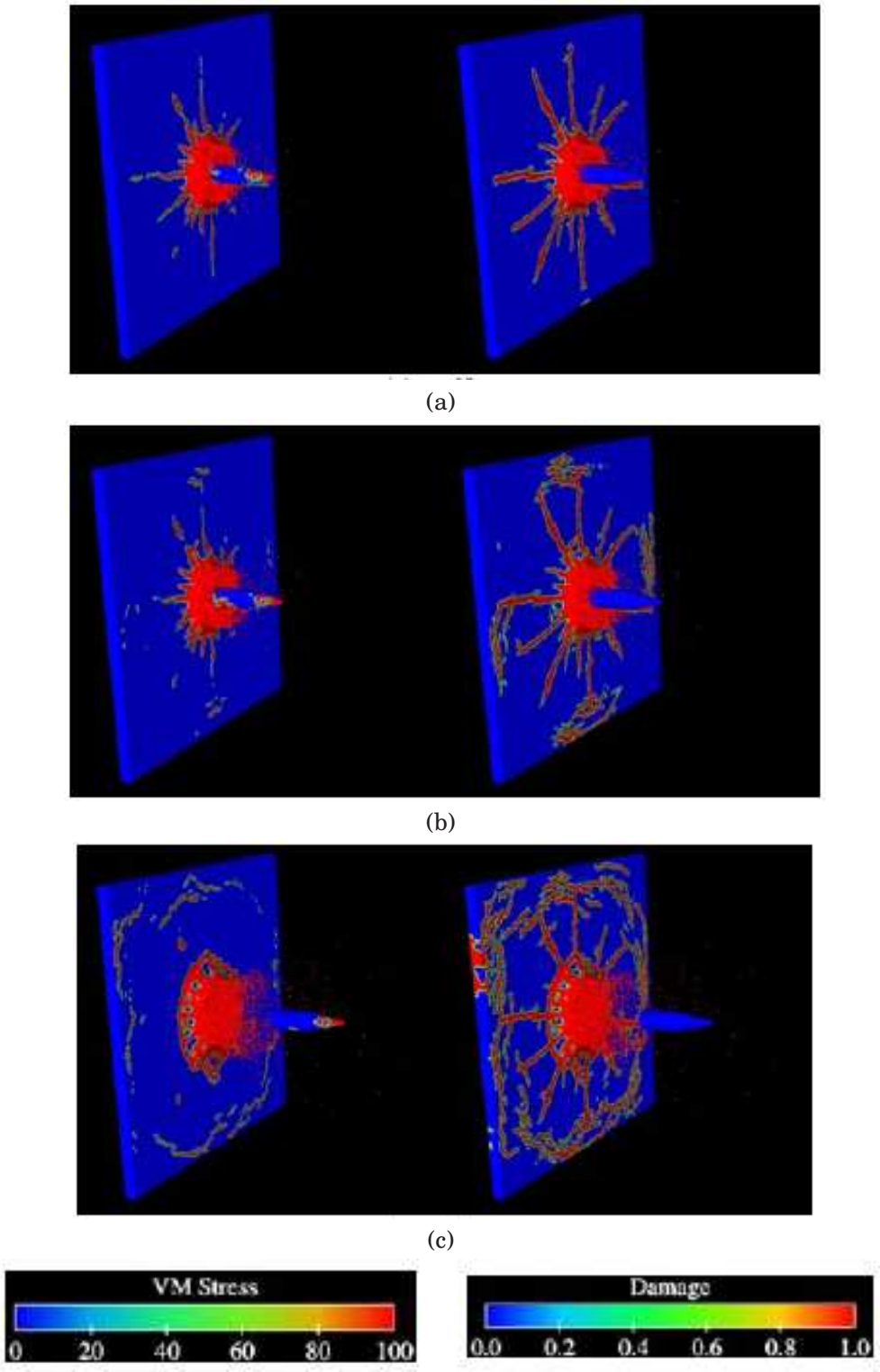
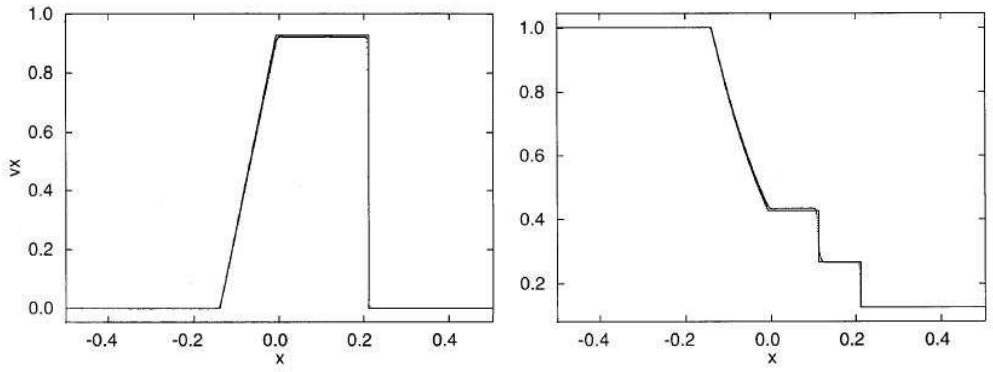


Figure 1.9: Fracture process for the plate during collision (left: colored by von Mises stress and right: colored by damage). (a) time $= 60\mu s$; (b) time $= 65\mu s$; (c) time $= 100\mu s$ [21].

1.4.3 SPH applications in fluid mechanics

Although the SPH method is still used for astrophysical simulations, its applications in fluid simulations are most notable. From compressible flows to incompressible flows, the SPH method shows reliable performance. In 2007, Monaghan [74] successfully simulated a shock tube with a SPH Riemann solver, and the density and velocity jump of the gas in the shock tubes were well predicted, shown in Fig. 1.10. Moreover, Welton [118] incorporated elements of SPH to extract mean quantities from the particles, including the mean pressure gradient, in simulations for compressible turbulent flows.



(a) Velocity against distance (b) Density against distance

Figure 1.10: Velocity and density profiles in shock tubes. Dot points : SPH simulations; lines: analytical results. [74]

Despite the good performance of the SPH method in compressible flow simulations, the SPH method reveals some difficulties in incompressible flow simulations, and this is now a main research focus. After the first application of the SPH method in incompressible flow simulations in [72], simulations of incompressible flows with the SPH method turned to be a critical area in CFD. Fig. 1.11 shows the simulations of the internal flow, free-surface flow and turbulence phenomena with the SPH method [47, 55, 117]. Compared to

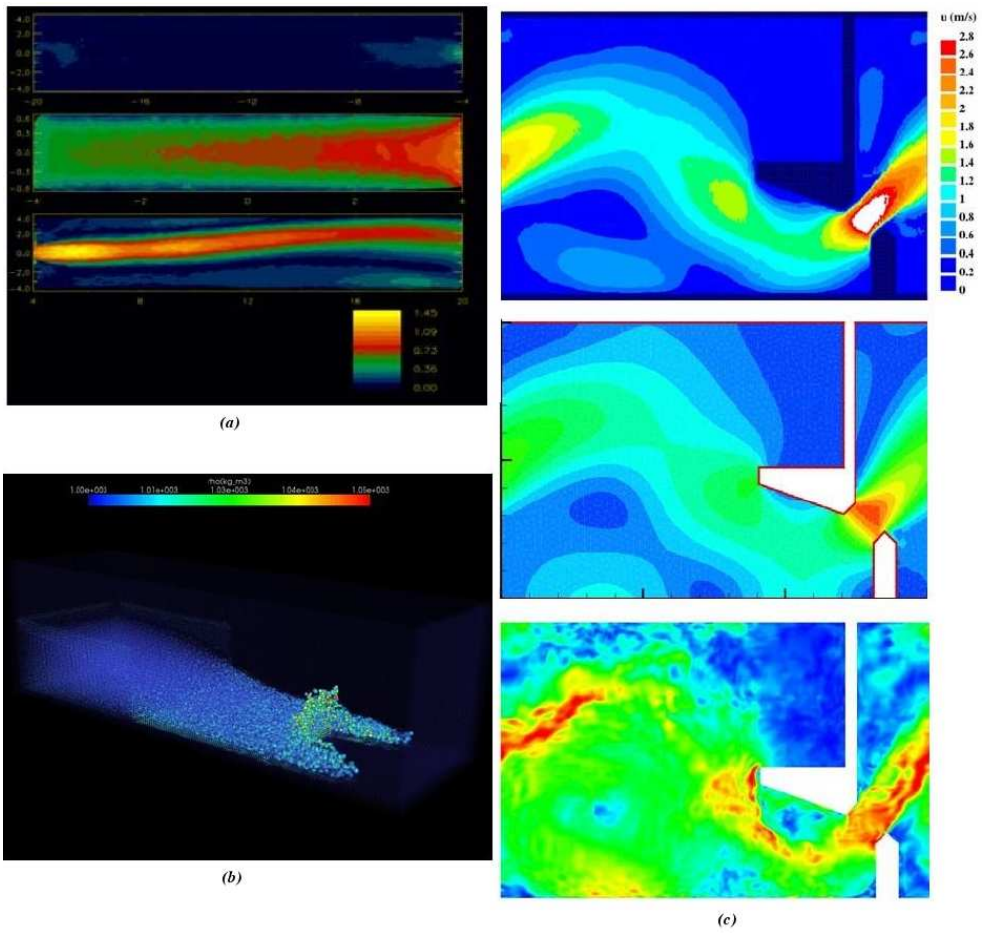


Figure 1.11: Simulations of incompressible flows by the SPH method. (a) flow through an orifice conduit [55]; (b) benchmark dam-break case with parallel computations [47]; (c) simulation of fish pass with turbulence models [117].

mesh-based methods, the meshless SPH method is more suitable for violent deformation, such as violent change of free surfaces. Also, similar to its vortex method counterpart, because of the Lagrangian movement of particles in SPH simulations, at least for approaches without the remeshing, or during processes before remeshing [25, 120], the SPH method is better suited to the prescription of the physical processes in the fluid.

SPH simulations of the incompressible flows can be performed by two methods: 1) approximately simulating incompressible flows with a small com-

pressibility, namely Weakly Compressible SPH (WCSPH); 2) simulating flows by enforcing incompressibility, namely Incompressible SPH (ISPH). In WCSPH method, the flow is considered as slightly compressible, with a state equation for the pressure calculation [18, 72, 77]. In ISPH methods the pressure-velocity coupling has been generally achieved by the projection method [12, 17, 44, 100, 63]. The recent alternative approach of imposing the kinematic constraint of a constant volume for each fluid particle through non-thermodynamic pressure is also potentially competitive as shown through comparisons with WCSPH in [25]. In this work, only the projection-based ISPH methods are studied.

When modelling incompressible flows, traditional SPH solvers resort to a weakly compressible approach. The incompressibility is enforced approximately by the use of a Mach number of approximately 0.1 [71]. It is easy to program because the pressure is given by an algebraic thermodynamic equation [72][77]. The weakly compressible approach to simulate incompressible flows causes problems, such as sound wave reflection at the boundaries and the high sound speed leading to a small time step resulting in high computing cost. Small fluctuations in fluid density cause big errors on the pressure field and numerical instability due to the use of the state equation [63].

Therefore, researchers are looking for better approaches to simulate incompressible flows. A different approach to model free surface, incompressible flows using a fully Lagrangian technique was the particle method proposed by Koshizuka et al. [56], where a penalty-like formulation was employed to adjust the pressure where density variations occurred. An iterative process that converged when density changes were below a specified tolerance was used. A similar approach was used in [58], where a pressure Poisson

equation was solved instead of a penalty-like method with source terms proportional to density variations. It is not clear though how efficient or accurate these methods are for free-surface modelling compared to the use of WCSPH. In 1999, Cummins and Rudman [17] applied a projection method in SPH, which projects an intermediate velocity field onto a divergence-free field and a curl-free field respectively. Shao and Lo [100] used an incompressible method, similar to that in [58], to describe the free surface in dam-break flow. Colin et al. [14] proposed an improved Laplacian operator in a method similar to [17]. In 2007 and 2008, Lee et al. [62, 63] pointed out that a truly ISPH method could improve the accuracy of the SPH method. In 2007, different from the projection-based ISPH, the non-thermodynamic pressure in [25] is calculated to maintain constant volume for each fluid particle. In 2007, Hu and Adams [44] proposed a stable algorithm ensuring both divergence-free velocity field and constant density, with an additional Poisson equation solution. A new ISPH algorithm is proposed in this thesis. The comparison of the new algorithm to the existing ones is conducted. The accuracy and stability of ISPH methods based on the projection algorithm is also discussed.

Another issue to investigate in this thesis is to use ISPH method to simulate incompressible free-surface flows. The ability of the SPH method to capture the fundamental fluid motion is needed in free-surface flow simulations. A mesh-based method, Volume of Fluid (VOF) method, was proposed by Hirt and Nichols [40], and it receives strong popularity nowadays in two-phase or free-surface flow simulations [20, 22, 60, 65]. But it has numerical difficulties in capturing the sharp interface with rapid changes [108]. As famous VOF researchers Hirt and Nichols [40] said, it is the flaw of the Eulerian approach, fluid elements are convected through the grid interface with the av-

eraged velocity instead of its own, which make it difficult to describe the free surface. This is the reason why Fox and McDonald [30], in their introductory fluid mechanics text book, state "Clearly the type of analysis depends on the problem. Where it is easy to keep track of identifiable elements of mass, we use a method of description that follows the particles".

In 1999, Monaghan used the Weakly Compressible SPH (WCSPH) method to simulate the solitary wave propagating along the beach [75]. Meanwhile, instead of simulating only one phase in two-phase flows, Colagrossi and Landrini [15], with WCSPH method, treated two-dimensional interfacial flows with different fluids separated by sharp interfaces. In 2004, Rogers et al. [96] used a combined Large Eddy Simulation (LES) type scheme or subparticle scale (SPS) scheme to capture coherent turbulent structures in breaking waves. Dalrymple and Rogers [18] applied a LES model to wave propagation and interaction with coastal defence. In 2007, Violeau and Issa [116] modeled the free-surface flow with the consideration of complex turbulence models. To better capture the physical process of the violent free-surface change, Ferrari et al. [28] performed simulations involving millions of particles, running on modern massively parallel supercomputers. Although WCSPH predicted some highly transient flows quite well, notably dam break flows, pressures were extremely noisy and the method highly dissipative. An important development was made by Vila [114] who introduced a Riemann formulation between interacting particles, reducing pressure noise markedly. It is however not clear here whether ISPH or Riemann-based SPH will finally win the competition on free-surface predictions.

The ISPH method with properties of noise-free pressure field and bigger time step limits is also preferable in free-surface flow simulations. In 2002 [66]

and 2003 [100], Lo and Shao applied a projection-based ISPH method in the simulation of free-surface flows. In 2004, Shao and Gotoh [101] simulated the coupled motion between progressive wave and floating curtain-wall type breakwater with consideration of turbulence effect through a Large Eddy Simulation (LES) model. The interface idea applied in WCSPH method in [15] was also adopted in the ISPH method by Hu and Adams [44]. In 2008, Lee et al. [63] applied a so-called Truly Incompressible SPH (TISPH) method to simulate a 2-D dam break. In [54], the ISPH method similar to the one introduced in [63] is also compared with the counterpart in [66, 100], with previous one showing obvious improvement over the later method. But the free-surface predictions are still noisy in [54]. The ISPH method is also applied here for free-surface flow simulations. The problem in ISPH method for free-surface prediction is first analyzed here, and one possible solution is proposed. Rigorous validation is conducted against analytical or highly accurate solutions as undertaken for internal flows through impulsive fluid-structure interaction, free-surface evolution with very high curvature and wave propagation.

1.5 Outlines of thesis

In this thesis, the basic SPH methodologies and incompressible SPH methods, including the new ISPH approach, will be introduced in Chapter 2. The structure of the incompressible SPH code will be presented in Chapter 3. In Chapter 4, the comparison between existing ISPH methods and the new ISPH method is conducted, and the accuracy of the new ISPH method will be investigated. In Chapter 5, the validation of the method and the code is

conducted through simulations of lid-driven cavity and flows around a circular bluff body in a channel. And the application of the new ISPH approach in free-surface flows is presented in Chapter 6. Finally, conclusions are drawn.

Chapter 2

The SPH Methodology, Existing and New ISPH Methods

In the SPH method, the Navier-Stokes equations are solved in Lagrangian form, defined by Eq. 2.1 and 2.2. Eq. 2.3 is also solved for the particle positions. SPH deals with operators for first and second derivative terms, and the pressure gradient as well as the viscous terms and Laplacian operators are presented in the following. Incompressibility here is enforced in the projection method by a pressure Poisson equation [12].

$$\nabla \cdot \mathbf{u} = 0 \tag{2.1}$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + \mathbf{F} \tag{2.2}$$

$$\frac{D\mathbf{r}}{Dt} = \mathbf{u} \tag{2.3}$$

where $\mathbf{u}$ is the particle velocity; P is the pressure; F is an extra body force; ν is the kinematic viscosity; ρ is the density; t is the time; $\mathbf{r}$ is the particle

position.

2.1 SPH methodology

2.1.1 Basic SPH methodology

For a variable A, in SPH formalism, the value of A at a point $\mathbf{r}$, with $\mathbf{r} = (x, y)$, is written as a convolution product of the variable A with the Dirac δ function

$$A(\mathbf{r}) = \int A(\mathbf{r}')\delta(\mid \mathbf{r} - \mathbf{r}' \mid)d\mathbf{r}'. \tag{2.4}$$

In SPH, the δ function is approximated by a kernel function $\omega_h(\mid \mathbf{r} - \mathbf{r}' \mid)$, with a smoothing length h. Therefore, Eq. 2.4 can be approximated as

$$A(\mathbf{r}) \approx \int_\Omega A(\mathbf{r}')\omega_h(\mid \mathbf{r} - \mathbf{r}' \mid)d\mathbf{r}', \tag{2.5}$$

where Ω is the supporting domain. In a discrete format, the interpolation can be written as

$$A(\mathbf{r}_\mathrm{i}) \approx \sum_j V_j A(\mathbf{r}_\mathrm{j})\omega_h(r_{ij}). \tag{2.6}$$

where V_j is the volume of particle j, and subscript j stands for the neighboring particle; r_{ij} is the distance between particle i and j. A quintic kernel is used for all the SPH interpolation in this thesis [77]. It is written as

$$\omega_{ij} = \frac{7}{478\pi} \begin{cases} (3-q)^5 - 6\,(2-q)^5 + 15\,(1-q)^5\,, & 0 \le q \le 1; \\ (3-q)^5 - 6\,(2-q)^5\,, & 1 \le q \le 2; \\ (3-q)^5, & 2 \le q \le 3; \\ 0, & q \ge 3, \end{cases} \tag{2.7}$$

where $q = |\mathbf{r}_{ij}|/l_s$. l_s is the kernel support size, $l_s = 3 \cdot h$, with $h = c \cdot \delta r$. δr is the initial particle distance, c is a constant, equal to 1.3 for all simulations in this thesis. Some other kernels can be found in Appendix A.

Straightforwardly, the gradient and divergence can be respectively written as

$$\nabla A(\mathbf{r}) \approx \int_\Omega A(\mathbf{r}')\nabla\omega_h(|\ \mathbf{r} - \mathbf{r}'\ |)d\mathbf{r}', \tag{2.8}$$

$$\nabla \cdot \mathbf{A}(\mathbf{r}) \approx \int_\Omega \mathbf{A}(\mathbf{r}') \cdot \nabla\omega_h(|\ \mathbf{r} - \mathbf{r}'\ |)d\mathbf{r}', \tag{2.9}$$

or

$$\nabla A(\mathbf{r}) \approx \sum_j V_j A(\mathbf{r}_\mathrm{j})\nabla\omega_h(r_{ij}), \tag{2.10}$$

$$\nabla \cdot \mathbf{A}(\mathbf{r}) \approx \sum_j V_j \mathbf{A}(\mathbf{r}_\mathrm{j}) \cdot \nabla\omega_h(r_{ij}). \tag{2.11}$$

These operator are referred as the basic gradient or divergence operator. In the later part, $\omega(r_{ij})$ and $\nabla\omega_h(r_{ij})$ are simply referred as ω_{ij} and $\nabla\omega_{ij}$ respectively.

2.1.2 SPH operators

Gradient and divergence operators

To solve the Navier-Stokes equations, the gradient and the divergence operators are needed. It has been shown in the previous part that the basic gradient or divergence operator can be obtained directly from SPH interpolation. However, it may result in large problematic errors [82]. Through simple algebraic manipulation, an expression for $\alpha^n \nabla A$, with integer $n \neq 0$, can be formulated as

$$\alpha^n \nabla A = \nabla \left(\alpha^n A\right) - n\alpha^{n-1} A \nabla \alpha \tag{2.12}$$

where α is a variant scalar. Substituting Eq. 2.12 with Eq. 2.10, the SPH formulation for gradient is

$$\alpha^n \nabla A \approx \sum_j V_j \left[\left(\alpha^n A\right)_j - \left(n\alpha^{n-1} A\right)_i \alpha_j\right] \nabla \omega_{\mathrm{ij}} \tag{2.13}$$

If $n = 0$, ∇A can be deduced through a similar simple algebraic operation as

$$\nabla A_i \approx \sum_j V_j (A_i + A_j) \nabla \omega_{ij} \tag{2.14}$$

$$\nabla A_i \approx -\sum_j V_j (A_i - A_j) \nabla \omega_{ij} \tag{2.15}$$

In [4], Bonet and Lok presented a correction method to preserve the angular momentum in gradient or divergence operators. In 2007, Oger et al. [82] analyzed the accuracy of different operators, and pointed out that with Eq. 2.15 the accuracy can be improved up to second order with the normaliza-

tion of the kernel function. The same normalization of the kernel was also proposed by Vila [114]. The expression for the kernel normalization can be expressed as

$$\nabla W_{ij} = \mathbf{L}(\mathbf{r}) \nabla \omega_{ij}. \tag{2.16}$$

where $\mathrm{r} = (x, y)$, ∇W_{ij} is substituted into Eq. 2.15 for $\nabla \omega_{ij}$ as a normalized kernel first derivative, and

$$\mathbf{L}(\mathbf{r}) = \begin{pmatrix} \sum\limits_j V_j \left(x_j - x\right) \dfrac{\partial \omega_{ij}}{\partial x} & \sum\limits_j V_j \left(x_j - x\right) \dfrac{\partial \omega_{ij}}{\partial y} \\ \sum\limits_j V_j \left(y_j - y\right) \dfrac{\partial \omega_{ij}}{\partial x} & \sum\limits_j V_j \left(y_j - y\right) \dfrac{\partial \omega_{ij}}{\partial y} \end{pmatrix}^{-1}. \tag{2.17}$$

A similar operator with kernel normalization is also used for divergence. The normalization cannot improve the accuracy of Eq. 2.14 to second-order as Eq. 2.15. This was first addressed in [82]. However, Eq. 2.14 has the antisymmetric character, the influence from particle i to particle j is the same magnitude as the influence from particle j to particle i, but with a opposite sign, which means Eq. 2.14 implies the Newton's third law (action-reaction principle), and the linear momentum is conserved. In [82], Oger et al. pointed out that although the conservation law is breached by Eq. 2.15 with the normalization, the high accuracy obtained by the normalization still places Eq. 2.15 in a superior position to Eq. 2.14.

Viscous term and Laplacian operator

In 1997, Morris et al. [77] deduced the laminar viscosity term by combining SPH and FDM approaches. The viscous term is presented in Eq. 2.18. In this approximated viscous term, only the first derivative of the kernel func-

tion is used. The calculation of first derivative can be used not only in the divergence or gradient calculation, but also for the viscous term. This saves the computing time. One of the viscous term expressions used in this work is identical to the one suggested by Morris [77],

$$(\nabla \cdot \mu \nabla \mathbf{u})_i \approx \sum_j \frac{m_j(\mu_i + \mu_j)\mathbf{r}_{ij} \cdot \nabla \omega_{ij}}{\rho_j(r_{ij}^2 + \eta^2)} \mathbf{u}_{ij} \tag{2.18}$$

where m is the mass of the particle; ρ is the fluid density; μ is the dynamic viscosity; $\mathbf{u}_i$ is the velocity of particle i, $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$; $\mathbf{r}_i$ is the position of particle i, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, and $r_{ij} =\mid \mathbf{r}_{ij} \mid$; η is a small value to avoid singular denominator. With constant viscosity and density in this thesis, a similar viscosity term is formulated as

$$(\nabla \cdot \mu \nabla \mathbf{u})_i \approx \sum_j \frac{2m_j\mu_j \mathbf{r}_{ij} \cdot \nabla \omega_{ij}}{\rho_j(r_{ij}^2 + \eta^2)} \mathbf{u}_{ij}. \tag{2.19}$$

An approximate Laplacian operator with the same format is used for scalars. Here is the example of the pressure.

$$\triangle p_i \approx \sum_j 2\frac{m_j}{\rho_j} \frac{p_{ij}\mathbf{r}_{ij} \cdot \nabla \omega_{ij}}{(r_{ij}^2 + \eta^2)} \tag{2.20}$$

where, $p_{ij} = p_i - p_j$. In [14], a new Laplacian operator was proposed, but with the calculation of the second derivative of the kernel, introducing extra computing expense. In [10, 11] Chen et al. by inverting the whole matrix for each particle, normally 3×3 for 2-D problem and 6×6 for 3-D problem, calculate the whole Hessian, $\frac{\partial^2 A}{\partial x_i \partial x_j}$, for any general variable A, and achieve 2nd-order accuracy. However, this demands a huge computing expense, for simulations with millions of particles almost practically impossible. In 2007,

Schwaiger [99] deduced another viscous/Laplacian operator with better performance around the free surface, shown in Eq. 2.21.

$$\begin{aligned}(\nabla \cdot \mu \nabla A)_i \;\approx\; & \frac{tr\left(\mathbf{\Gamma}^{-1}\right)}{n}\{\sum_j V_j(\lambda_j+\lambda_i)\left(A_j-A_i\right)\frac{\mathbf{r}_{ij}\cdot\nabla\omega_{ij}}{r_{ij}^2} \\ - \;& \left[\nabla\left(\lambda_i A_i\right)-A_i\nabla\lambda_i+\lambda_i\nabla A_i\right]\cdot\left(\sum_j V_j\nabla\omega_{ij}\right)\}\end{aligned} \tag{2.21}$$

where $\mathbf{\Gamma}$ is a tensor, and written as

$$\Gamma_{\alpha\beta}=\sum_j \frac{\mathbf{r_{ij}}\cdot\nabla\omega_{ij}}{r_{ij}^2}\left(r_{ij}\right)_\alpha\left(r_{ij}\right)_\beta . \tag{2.22}$$

And α and β are the coordinate directions. λ is the diffusion coefficient, which is assumed to be spatially variant. The Laplacian operator can be obtained by simply setting $\lambda = 1$. The influence of two different Laplacian operators on the accuracy and spatial convergence speed is discussed in Chapter 5, and their performances in free-surface cases with the truncated kernel are also compared in Chapter 6.

2.1.3 Wall boundary conditions

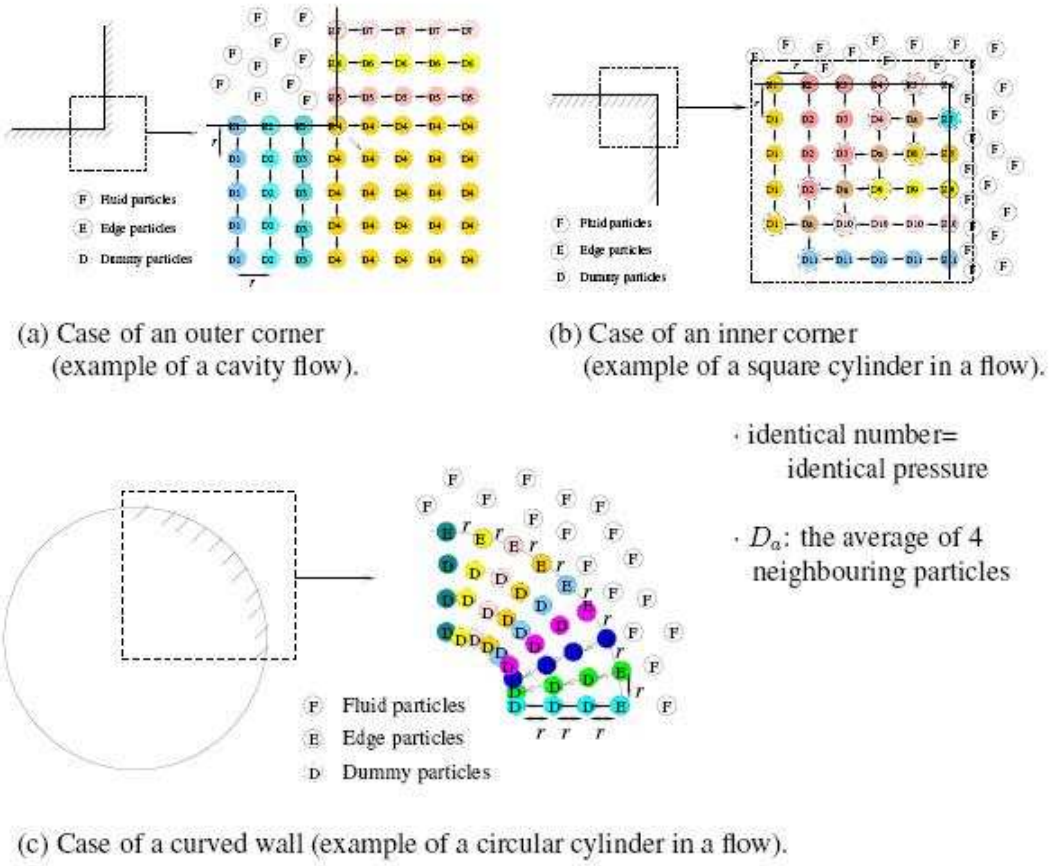
Since the first time the SPH method was proposed, three widely-used wall boundary conditions have been used, the dummy particle method [63, 100], the mirror particle method [77, 72] and the boundary force method [18]. It is very difficult to build some boundary conditions, such as the homogeneous Neumann boundary for the pressure, with the boundary force method. Therefore, the dummy particle and the mirror particle methods are normally used in incompressible SPH simulations. Both of the boundary methods

have been applied in the incompressible SPH code. The comparison has been made, presented in the next chapter about the code structure.

Dummy particles and wall boundaries

In the dummy particle method, several extra layers of particles are placed inside the wall, with the same pressure and velocity values as corresponding normal direction solid wall particles. The dummy particle method is applied in the incompressible SPH method successfully by Lee et al. in [62]. The special treatment for the inner or outer corners and curved surfaces in [62], is shown in Fig. 2.1. For the outer corner, dummy particles, D2 in Fig. 2.1 (a), carry the identical velocity and pressure value to their wall particle, W2 in Fig. 2.1 (a). For the inner corner, dummy particles for the two walls, which are perpendicular to each other, overlap with the same treatment for the flat wall. Special treatment is done by using the averaged value for dummy particles at the diagonal position, Da in Fig. 2.1 (b). For the curved wall, dummy particles are located along the opposite direction of the wall normal.

Dummy particles are easy to implement. They are set up at the first time step, without update as the time marches, which is not the case in mirror particle method. The discussion about wall boundaries with dummy particle method, especially about the inner/outer corner and curved wall, will not be extended here. More detail about the dummy particle method can be found in [63, 100]. Dummy particles are only used for the flat wall boundary in this thesis in comparison with the mirror-particle method in §3.5.



(a) Case of an outer corner (example of a cavity flow).

(b) Case of an inner corner (example of a square cylinder in a flow).

(c) Case of a curved wall (example of a circular cylinder in a flow).

Figure 2.1: The special treatment for dummy particles at the inner or outer corner and the curved wall. Blue circles: fluid particles; red circles: wall particles; yellow circles: dummy particles; purple circles: dummy particles for the special treatment at corners.

Mirror particles and wall boundaries

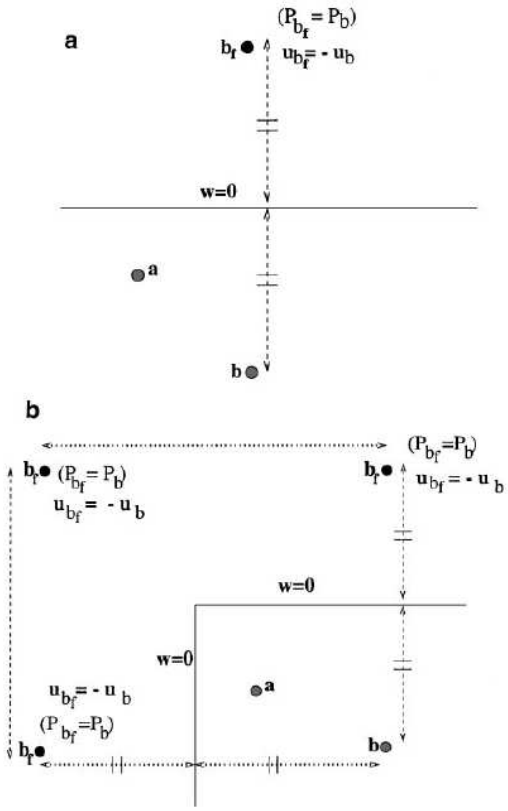


Figure 2.2: Mirror particle method for the static wall boundary. u_f and u_m are the velocities for the fluid particle and its corresponding mirror particle respectively; p_f and p_m are the pressures for the fluid particle and its corresponding mirror particle respectively; w is the wall velocity, and here $w = 0$.

The mirror particle method is firstly proposed by Morris et al. in 1997 [77] to build the non-slip boundary condition at walls. Cummins and Rudman [17] used the mirror particle method to build both non-slip and Neumann boundaries for walls. The particles close to the wall, whose distances to the wall are less than the support size, are reflected on the other side of the wall as mirror particles or fictitious particles. Mirror or fictitious particles carry the same pressure as their corresponding fluid particles, and their velocity

values are calculated by the central differencing scheme,

$$\mathbf{w} = (\mathbf{u}_{if} + \mathbf{u}_i)/2 \tag{2.23}$$

where $\mathbf{u}_i$ and $\mathbf{u}_{if}$ are the velocities for fluid particle i and its corresponding mirror particle respectively, w is the wall velocity. At a corner, the special treatment, similar to the counterpart in the dummy particle method, is performed, shown in Fig. 2.2.

2.1.4 Instability in SPH

In [5, 27, 44], it was shown that increasingly irregular particle distributions exhibit increasing numerical errors in results. Fang and Parriaux [27] pointed out that an ill-conditioned matrix in the linear system could appear, with increasing non-uniformity of particle distribution.

In [76], Monaghan pointed out that the tensile instability in SPH results in the clustering of particles. The clustering is particularly noticeable in materials which have an equation of state which can give rise to negative pressures, but it can occur in the fluid where the pressure is always positive. It is a particular problem in solid body computations where the instability may corrupt physical fragmentation by numerical fragmentation which, in some cases, is so severe that the dynamics of the system is completely wrong.

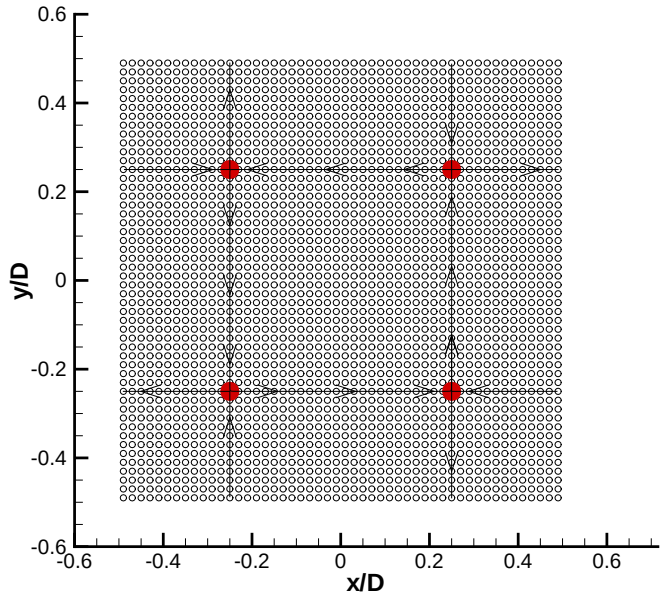


Figure 2.3: Streamlines in Taylor-Green vortices flow. The clustering points are marked by red points. Coordinates have been non-dimentionalized by the unit square length D. $\circ$ = fluid particles; $\longrightarrow$ = streamlines.

Here, the author takes this instability problem as the effect of the kernel flaw. The non-uniformity or the clustering could reasonably happen in some special fluid situations, taking Taylor-Green flow as an example. When the Navier-Stokes equations are solved in Lagrangian form, fluid particles move along stream lines. Therefore, it is very possible that fluid particles clump together at stagnation points in the fluid field, shown in Fig. 2.3. However, the flaw of the smoothing kernel introduces interpolation error. Fig. 2.4 (a) shows the contour graph of the 2-D quintic kernel value [77], and 2.4 (b) its first derivative. From the cross-section profile of the first derivative, at $x = 0$, Fig. 2.4 (c), it can be seen that when particles are getting close to each other within a certain distance range, the interaction between them is not increased, but reduced. This non-physical behavior of the kernel func-

tion introduces huge error in simulations, such as in the pressure gradient, or the Laplacian operator, and causes the particle clustering phenomenon, resulting in non-physical predictions.

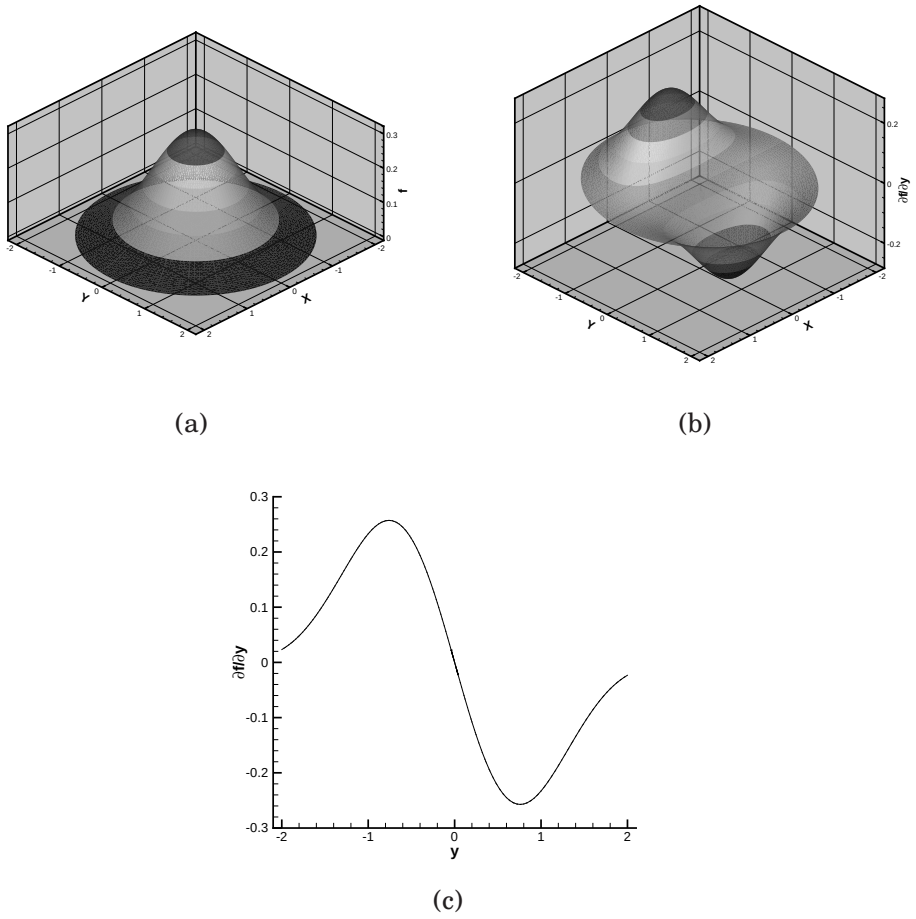


Figure 2.4: (a) contour graph for 2-D quintic kernel; (b) contour graph for the first derivative, $\frac{\partial f}{\partial y}$, of 2-D quintic kernel; (c) profile for the first derivative of quintic kernel, $\frac{\partial f}{\partial y}$, at cross section $x = 0$.

Although it is reported that the quadratic kernel can reduce this non-physical behavior [18], its low-order accuracy and second derivative property, which strongly influences the stability of the SPH method [77], limits its application in high-accuracy simulations. In [76], Monaghan introduced an extra artificial stress in the pressure term to overcome particle clustering problems.

An effective alternative is to remesh on a fixed uniform grid as proposed by Chaniotis et al. [9]. In this thesis, a new method, based on the projection method with slight particle shifting, is introduced to overcome particle clustering without relying on the background uniform mesh**.** The shifting method is first introduced by Nestor et al. [80] in the context of Finite Volume Particle Method (FVPM), where the particles are shifted based on the local particle distribution.

2.2 Existing projection-based ISPH method

2.2.1 ISPH with divergence-free velocity field (ISPH_DF)

To keep a divergence-free velocity field, the projection method [12] is used, as first presented by Cummins and Rudman [17] in 1999. This incompressible divergence-free SPH method will be referred to as ISPH_DF. A second-order time marching scheme is applied, where both the density and mass of particles are constant. Particle positions, $\mathbf{r}_i^n$, are updated with velocity $\mathbf{u}_i^n$ to positions $\mathbf{r}_i^*$,

$$\mathbf{r}_i^* = \mathbf{r}_i^n + \triangle t \mathbf{u}_i^n \tag{2.24}$$

where $\triangle t$ is the time step. An intermediate velocity $\mathbf{u}_i^*$ is calculated at the position, $\mathbf{r}_i^*$, based on the momentum equation without the pressure gradient term,

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \left(\nu \nabla^2 \mathbf{u}_i^n + \mathbf{F}_i^n\right) \triangle t, \tag{2.25}$$

where, $\mathbf{F}_i^n$ is a body force. This intermediate velocity $\mathbf{u}_i^*$ can be projected onto a divergence-free velocity field and a curl-free pressure gradient field [12] as

$$\mathbf{u}_i^* = \mathbf{u}_i^{n+1} + \frac{\triangle t}{\rho}\nabla p_i^{n+1}. \tag{2.26}$$

By taking the divergence of Eq. 2.26 and enforcing $\nabla \cdot \mathbf{u}_i^{n+1} = 0$, the pressure at time $n+1$ can be obtained from the pressure Poisson equation (PPE), written as

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p^{n+1}\right)_i = \frac{1}{\triangle t}\nabla \cdot \mathbf{u}_i^*. \tag{2.27}$$

If there is a free surface, free-surface particles need to be identified with the criterion described in §6.1 in Chapter 6 and the pressure values for free-surface particles are forced to be 0 in the pressure Poisson equation, Eq. 2.27. The velocity at time $n+1$, $\mathbf{u}_i^{n+1}$ will result from the projection of $\mathbf{u}_i^*$, Eq. 2.26. Therefore,

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\triangle t}{\rho}\nabla p_i^{n+1}. \tag{2.28}$$

The particle position is finally advanced in time,

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \triangle t\left(\frac{\mathbf{u}_i^{n+1} + \mathbf{u}_i^n}{2}\right). \tag{2.29}$$

In an alternative formulation, Lee et al. [63] derived the velocity from the pressure correction with the particles at $\mathbf{r}_i^n$ before advancing their positions with the total velocity, termed as truly incompressible SPH. This formulation is also applied here and produces results identical to [17] for the test cases investigated. An interesting point is that Eq. 2.27 is obtained from the continuity equation, $\nabla \cdot \mathbf{u} = 0$, therefore, the divergence operator in the Laplacian

should be the same as the one used in the velocity divergence calculation, and the gradient operator should be the same as the pressure gradient correction part in Eq. 2.28, which means the compatibility for the operators. However, Cummins and Rudman [17] proved that the compatible Laplacian operator causes the checker-board effect in the pressure field, similar to the one observed with collocated Finite Volume Method [93].

2.2.2 ISPH with density invariance (ISPH_DI)

Similar to the method in [58], Shao and Lo [100] proposed a projection-based incompressible method to impose density invariance to describe free-surface flows. This density-invariant ISPH method is referred to as ISPH_DI. In this case, the density is not kept constant in the course of the simulation. An intermediate velocity, $\mathbf{u}_i^*$, is calculated without the pressure gradient term as before,

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \left(\nu\nabla^2\mathbf{u}_i^n + \mathbf{F}_i^n\right)\triangle t. \tag{2.30}$$

The particle positions are updated to an intermediate position, $\mathbf{r}_i^*$,

$$\mathbf{r}_i^* = \mathbf{r}_i^n + \triangle t\mathbf{u}_i^*. \tag{2.31}$$

At this intermediate position, the intermediate velocity is projected onto two spaces, ∇p^{n+1} and $\mathbf{u}_i^{n+1}$, which is similar to ISPH_DF. But instead of calculating the pressure field p^{n+1} through a Poisson equation with a velocity divergence on the right hand side (R.H.S.), the pressure field is obtained by solving a Poisson equation with a relative density difference on R.H.S., as

shown in Eq. 2.32.

$$\nabla \cdot \left(\frac{1}{\rho^*} \nabla p^{n+1} \right) = \frac{\rho_0 - \rho^*}{\rho_0 \triangle t^2} \tag{2.32}$$

where ρ_0 and ρ^* are respectively the initial and temporal fluid density of each particle, calculated as

$$\rho_i = \sum_j m_j \omega_{ij}. \tag{2.33}$$

And the intermediate velocity field is corrected by the pressure gradient,

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\triangle t}{\rho^*} \nabla p^{n+1}. \tag{2.34}$$

To complete the time step, as for ISPH_DF, a second-order time marching scheme is used.

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \triangle t \left(\frac{\mathbf{u}_i^{n+1} + \mathbf{u}_i^n}{2} \right) \tag{2.35}$$

Eq. 2.32 and Eq. 2.27 are actually equivalent, if one assumes the continuity equation for the flow to be valid at the intermediate position, $\mathbf{r}_i^*$, with

$$(\nabla \cdot \mathbf{u})_{\mathbf{r}_i^*} = - \left(\frac{1}{\rho} \frac{d\rho}{dt} \right)_{\mathbf{r}_i^*}. \tag{2.36}$$

Eq. 2.32 can be obtained by substituting Eq. 2.36 into Eq. 2.27. It will be shown in §4.1 and 4.2 in Chapter 4 that the numerical performance of both Poisson equations, Eq. 2.27 and Eq. 2.32, are quite different.

2.2.3 ISPH with velocity and density control (ISPH_DFDI)

In [44], Hu and Adams point out that if only a divergence-free velocity field is enforced, a large density variation will appear. They suggested that a

divergence-free and density-invariance algorithm should be applied. This requires two pressure Poisson equations to be solved in this algorithm. This combined incompressible SPH method is referred to as ISPH_DFDI.

Similar to Shao and Lo's method [100], to keep the density constant, represented as $\rho_i^n = \rho_i^0$, the particle positions are adjusted. First, intermediate velocities and particle positions are obtained by

$$\mathbf{u}_i^{*,n+1/2} = \mathbf{u}_i^n + \left(\nu\nabla^2\mathbf{u}_i^n + \mathbf{F}_i^n\right)\frac{\triangle t}{2} \tag{2.37}$$

$$\mathbf{r}_i^{*,n+1} = \mathbf{r}_i^n + \mathbf{u}_i^{*,n+1/2}\triangle t \tag{2.38}$$

The intermediate particle density $\rho^{*,n+1}$is calculated. A pressure Poisson equation, as presented by Eq. 2.39, is repeatedly solved through internal iteration to obtain the pressure for particle position adjustment.

$$\frac{\triangle t^2}{2}\nabla\cdot\left(\left(\frac{1}{\rho}\nabla p\right)^*_i\right) = \frac{\rho_i^n - \rho_i^{*,n+1}}{\rho_i^n} \tag{2.39}$$

where $\left(\frac{1}{\rho}\nabla p\right)^*_i$ is the intermediate pressure gradient to adjust the particle position, with $\rho_i = \rho_i^0$. If we define

$$\sigma_i = \sum_j \omega_{ij} \tag{2.40}$$

then

$$\rho_i = \sum_j m_j\omega_{ij} = m_i\sigma_i. \tag{2.41}$$

With the relation $\rho_i^n = \rho_i^0 = m_i\sigma_i^0$ and $\rho_i^{*,n+1} = m_i\sigma_i^{*,n+1}$, Eq. 2.39 can be written as

$$\frac{\triangle t^2}{2}\nabla\cdot\left(\left(\frac{1}{\rho}\nabla p\right)^*_i\right) = \frac{\sigma_i^0 - \sigma_i^{*,n+1}}{\sigma_i^0}. \tag{2.42}$$

After the pressure is calculated, the particle positions are adjusted through the pressure gradient $\left(\frac{1}{\rho}\nabla p\right)^*_i$ following

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^{*,n+1} - \left(\frac{1}{\rho}\nabla p\right)^*_i \frac{\triangle t^2}{2} \tag{2.43}$$

During the simulation, the particle density will be recalculated at position $\mathbf{r}^{n+1}$ from Eq. 2.41. A criterion for the relative density difference, typically 1%, is set. If the criterion is not fulfilled, particle positions will be adjusted by the following internal iteration.

$$\begin{gathered}\frac{\triangle t^2}{2}\nabla\cdot\left(\left(\frac{1}{\rho}\nabla p\right)^{*,m}_i\right) = \frac{\sigma_i^0 - \sigma_{i*}^{n+1,m}}{\sigma_i^0} \rightarrow \left(\frac{1}{\rho}\nabla p\right)^{*,m}_i \\ \mathbf{r}_i^{n+1,m+1} = \mathbf{r}_i^{n+1,m} - \left(\frac{1}{\rho}\nabla p\right)^{*,m}_i \frac{\triangle t^2}{2} \\ \sigma_{i*}^{n+1,m+1} \leftarrow \mathbf{r}_i^{n+1,m+1}\end{gathered} \tag{2.44}$$

where m is the number of internal iterations.

As for ISPH_DF, a Poisson equation needs to be solved to keep the velocity field divergence-free. First, the velocity field is explicitly calculated without considering the pressure gradient term.

$$\mathbf{u}_i^{*,n+1} = \mathbf{u}_i^{*,n+1/2} + \left(\nu\nabla^2\mathbf{u}_i^{*,n+1/2} + \mathbf{F}_i^{*,n+1/2}\right)\frac{\triangle t}{2}. \tag{2.45}$$

The velocity field is then corrected by the pressure gradient,

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^{*,n+1} - \frac{\triangle t}{2\rho}\nabla p_i^{n+1} \tag{2.46}$$

where, the pressure field is obtained from the pressure Poisson equation, shown as

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p^{n+1}\right)_i = \frac{2}{\triangle t}\nabla \cdot \mathbf{u}_i^{*,n+1}, \tag{2.47}$$

with $\rho_i = \rho_i^0$. An interesting point for this method is that the pressure and corresponding gradient fields are almost doubled due to the half-time-step operation. In the second half step, the value of $\mathbf{u}_i^{*,n+1}$ is actually the same as that in ISPH_DF. Comparing Eq. 2.27 and 2.47, the R.H.S., is doubled in Eq. 2.47. Therefore, the pressure and corresponding gradient fields are doubled in ISPH_DFDI, which will be shown in §4.2 in Chapter 4. A noteworthy point is that the pressure is only used as an internal mechanism to drive the velocity field in the correction stage, which happens in the half of the time step, as shown in Eq. 2.46. Herewith, the doubled pressure field does not influence the velocity results. Similar to Eq. 2.32 in ISPH_DI, Eq. 2.39 or 2.42 is solved to prevent particle clustering, while Eq. 2.47 is used to keep a divergence-free velocity field. Uniform particle spacings improve the stability of simulations. Through solving two Poisson equations, Eq. 2.39 and 2.47, both the density invariance and divergence-free velocity field are approximated.

2.3 New ISPH approach

In ISPH_DF, particles move along streamlines when the Lagrangian-form Navier-Stokes equations are solved accurately. The stretching and compressing of particle spacings, similar to the mesh distortion in the FV method, happen under certain conditions, highlighted by the Taylor-Green vortices, as shown in Fig. 2.5 (a). It is observed in the simulations that this non-uniform particle distribution hinders the convergence of linear solvers. Also, on the one hand, if the streamlines direct the particles towards each other, taking Taylor-Green vortices as an example, Fig. 2.3, the particles will cluster; on the other hand, the non-physical behavior of the kernel, introduced in Section 2.3, will weaken the interaction between particles when particle distances are within a close range. Then particles will continue clustering if the inertial motion is big enough, which is quantified by the Reynolds number, and

$$Re = \frac{UD}{\nu} \tag{2.48}$$

where U is the characteristic velocity of the flow; D is the characteristic length of the geometry; ν is the kinematic viscosity. The error, caused by this kernel flaw, also causes the instability in the projection-based ISPH.

The following method stabilizes the accurate ISPH_DF method. The particle distribution is effectively well maintained, shown in Fig. 2.5 (b). The pressure field is calculated as in ISPH_DF. And the particles are advanced, shifted slightly, and accordingly, the hydrodynamic variables corrected by the Taylor expansion,

$$\phi_{i'} = \phi_i + (\nabla\phi)_i \cdot \delta\mathbf{r}_{ii'} + \mathcal{O}(\delta r_{ii'}^2) \tag{2.49}$$

where, ϕ is a general variable; i and $i^{'}$ are the particle's old and new position respectively; $\delta \mathbf{r}_{ii'}$ is the distance vector between the particle's new and old position. The method with particle shifting is called ISPH_DFS. Here, only the first two terms are applied in simulations, giving an order consistent with the Laplacian operator. Higher order accuracy may be achieved with additional terms. With the position shifting and interpolation, the particles can jump from one streamline to another, and the particle clustering can be avoided; the error caused by highly-distorted particle spacings is reduced.

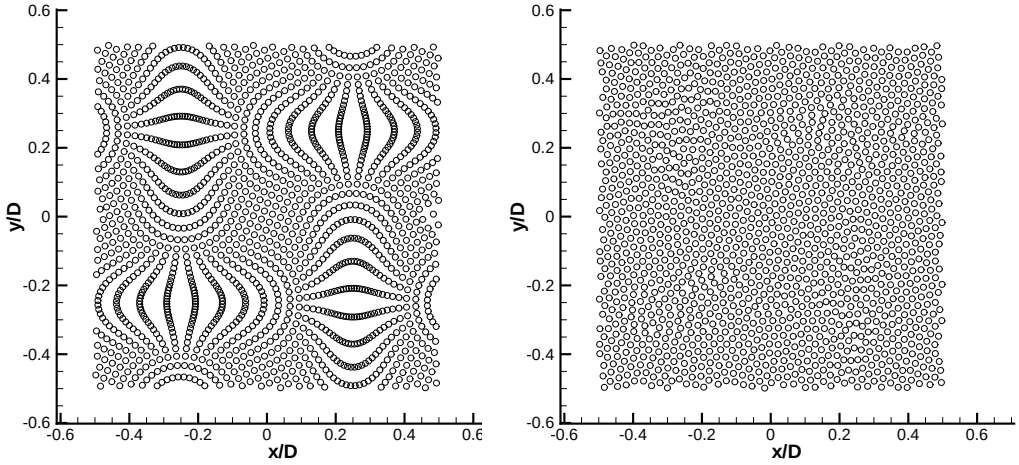


Figure 2.5: Particle distributions in simulations of Taylor-Green vortices with different methods, at $Re = 1000$. (a) Highly distorted particle spacings with ISPH_DF; (b) Well-maintained particle spacings with ISPH_DFS. ○ = fluid particles.

The idea of shifting particle positions has been proposed previously in the context of the Finite Volume Particle Method (FVPM) by Nestor et al. [80]. This allows the use of particle velocities with a correction, u′, added to conserve fluid momentum. Different from FVPM, the adjustment of particle

velocity here introduces sink or source terms to the total momentum. The position shift is applied to particles, and the use of a second-order Taylor expansion interpolates hydrodynamic variable values at the new position, as given in Eq. 2.49. Similar to the velocity correction equation, $\mathbf{u}'$, in [80], but modifying the particle shifting magnitude, α, in relation to the particle convection distance and the particle size, the position shift reads

$$\delta \mathbf{r}_i = C\alpha \mathbf{R}_i \tag{2.50}$$

where C is a constant, set as $0.01 - 0.1$, as discussed below; α is the shifting magnitude which is equal to the maximum particle convection distance $U_{max}\triangle t$, with U_{max} the maximum particle velocity, and $\triangle t$ the time step; $\mathbf{R}_i$ is the shifting vector, and reads:

$$\mathbf{R}_i = \sum_{j=1}^{M_i} \frac{\bar{r}_i^2}{r_{ij}^2} \mathbf{n}_{ij} \tag{2.51}$$

where, M_i is the number of neighboring particles around particle i, it is the same as the kernel interpolation for particles inside of the fluid, while for particles i close to the free surface, only particles which have smaller distance to the particle i than the distance of particle i to the free surface are considered; r_{ij} is the distance between particle i and particle j; $\bar{r}_i$ is the average particle spacing in the neighborhood of i, and

$$\bar{r}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} r_{ij}; \tag{2.52}$$

$\mathbf{n}_{ij}$ is the unit distance vector between particles i and j. The number of neighboring particles M_i is the same as that in the kernel interpolation for

particles inside of the fluid, while for particles i close to the free surface, only particles which have smaller distance to the particle i than the distance of particle i to the free surface are considered. The summation of $\mathbf{n}_{ij}$ actually represents the anisotropy of the particle spacings. $\bar{r}_i^2/r_{ij}^2$ is used here as a weighting function to reduce the influence from remote neighboring particles. R_i is evaluated on a fixed particle map obtained after the evolution equation Eq. 2.29. No internal iteration for a converged particle position is needed here.

The shifting distance should be large enough to prevent instability and small enough not to cause inaccuracy due to the Taylor expansion correction. Values of C within the range 0.01 - 0.1 satisfy these criteria for these test cases and the dependence on Reynolds number was not observed**.** A value of 0.04 is generally used here. In [80], an average particle spacing, as defined in Eq. 2.52, is used as the shifting magnitude α, considering the influence of particle size. However, it is observed in simulations that when shifting distances are much larger than convection distances, large numerical error appears in the fields of hydrodynamic variables, even with the Taylor expansion update. To avoid this an upper limit on shifting distance α is simply set as $U_{max}\triangle t$, updated at each time step. Note also that the shifting distance is always much less than the smoothing length h.

All in all, the algorithm can be summarized here as

- Convect particle i to an intermediate position $\mathbf{r}_i^*$, as shown in Eq. 2.24;
- Calculate an intermediate velocity, $\mathbf{u}_i^*$, without the pressure gradient term, as shown in Eq. 2.25;

- Calculate the pressure from the Poisson equation, as shown in Eq. 2.27;
- Correct the intermediate velocity, $\mathbf{u}_i^*$, by the pressure gradient to obtain the velocity for next time step, $\mathbf{u}_i^{n+1}$, shown in Eq. 2.28;
- The particle position is advanced in time, as shown in Eq. 2.29;
- Shift the particle by Eq. 2.50;
- Correct the velocity field by Eq. 2.49;
- Continue the calculation for next time step.

This small change in ISPH_DF makes this method much more robust, and the efficiency of simulations is improved in comparison with ISPH_DFDI. It should be pointed out here that it is not a strictly conservative method, similar to LFPM [27]. However it will be shown in the test cases in§4.3.3 that accuracy is not sacrificed, using first-order accurate interpolation, within the resulting stable algorithms.

2.4 Partial conclusion

The SPH method is a fully Lagrangian method. By approximating the Dirac δ function with a so-called smoothing function, the Navier-Stokes equations can be discretized on a meshless basis. However, by calculating the pressure from a state equation, the traditional WCSPH method cannot provide smooth prediction for the hydrodynamic field [63]. By applying the projection algorithm in SPH, the pressure field is smoother [63]. However, the

instability may occur due to highly distorted particle spacings. It is partially concluded here:

- The wall boundary condition is generally built up with mirror particle or dummy particle method. The dummy particle method is more efficient without updating the dummy particles during the simulations. Full comparison of these two boundary methods will be presented in the next chapter.

- By projecting the intermediate velocity field onto the divergence-free and curl-free fields, the projection algorithm highly improves the accuracy of the SPH method [63], taking the original incompressible SPH method presented by Cummins and Rudman [17] as an example. However, the instability is set in the simulations when the particle spacing is highly distorted. Another approach, called ISPH_DI here, was introduced by Shao and Lo [100], and the incompressibility is achieved by maintaining the particle volume constant. In 2007, Hu and Adams proposed another ISPH method, called ISPH_DFDI here, in which both constant particle volume and divergence-free velocity field are kept, and the stability is well kept by this method. In this chapter, a new approach called ISPH_DFS is also proposed. By slightly shifting the particle position and correcting the corresponding hydrodynamic field, the instability problem is overcome with ISPH_DFS method.

Chapter 3

The Incompressible SPH Code

THe incompressible SPH program is coded based on the original Weakly Compressible SPH (WCSPH) code SPHysics [121]. In SPHysics, the pressure field is calculated from a state equation without solving the pressure Poisson equation. This code is an open-source SPH code that has been released in 1 Aug 2007, developed jointly by researchers at the Johns Hopkins University (U.S.A.), the University of Vigo (Spain), the University of Manchester (U.K.) and the University of Rome La Sapienza (Italy) [121]. Version 1.4 and below contains the basic SPH formulation for free-surface flows. The 2-D & 3-D code has been developed specifically for free-surface hydrodynamics. The structure of the weakly compressible code SPHysics can be referred to Appendix A and [31].

Based on the code SPHysics, an ISPH code is programmed. Both Laplacian operators shown in §2.1.2 are built. Two linear solvers, namely Conjugated Gradient (CG) and Bi-Conjugated Gradient (Bi-CGSTAB), are programmed. The predictor-corrector time marching scheme is applied.

3.1 Linked list

In SPH simulations, the influence of neighbouring particles within the kernel support domain is considered. For a particular particle a, by calculating the particle distance to all the other particles by which the status, inside or outside of the support domain, is decided, the operations will be $(N-1)\cdot N$ for a N particles case. In the code SPHysics, square cells with a edge size equal to the kernel support size are placed in the simulation domain [112]. For each particle, the linked list is built up with the consideration of particles in neighbouring cells, which highly reduced the searching time. It has been proved that this method can reduce the operation number from $(N-1)\cdot N$ to $NlogN$ for N particles [70].

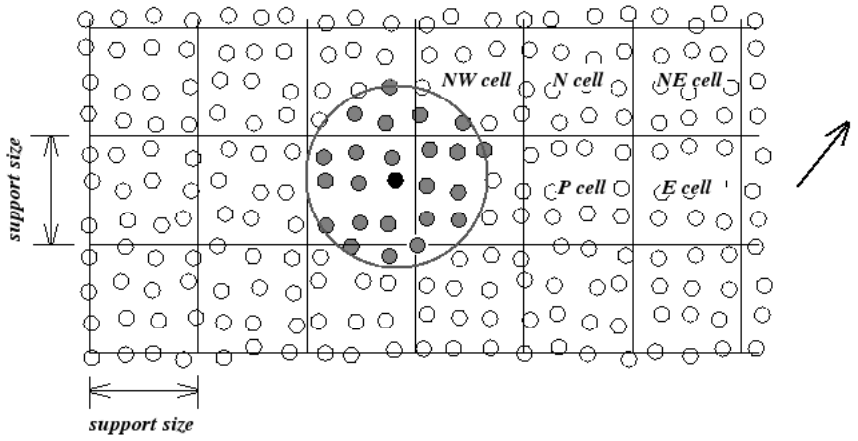


Figure 3.1: Neighbouring list in the incompressible SPH code. The neighbouring particles are those located in the adjacent cells. In 2-D, only NW, N, NE and E cells are used for neighbouring particle searching.

Another trick to speedup the building of the linked list is to search only particles in northwest (NW), north (N), northeast (NE), east (E) and polar (P) cells, shown in Fig.3.1. As shown in Chapter 2, all the SPH operators are symmetric or anti-symmetric. The calculation for different operator is only

needed once for each pair of particles. The influences of particles in west (W), southwest (SW), south (S) and southeast (SE) cells to particles in P cell have been considered when building linked list for particles in W, SW, S and SE cells. By only sweeping through the NW, N, NE and E cells, the computing cost is minimized.

A similar linked list building procedure can be easily extended to 3-D. Different linked list techniques can be found in [113].

3.2 Discretized Navier-Stokes equations and time marching

ISPH_DF algorithm in §2.2.1 in Chapter 2 is taken as an example for the illustration of the discretisation in the code. The simplest viscosity term, divergence, gradient and Laplacian operators are considered. Other expressions about algorithms and operators can be easily derived.

3.2.1 Calculation of intermediate velocity (u^*)

After particles are moved to the intermediate position r^* by Eq. 2.24 in Chapter 2, an intermediate velocity field $\mathbf{u}^*$ is calculated as shown in Eq. 2.25 in Chapter 2. If the viscous operator in [77] is used, $\mathbf{u}^*$ is written as

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\triangle t} \simeq \sum_j \frac{4m_j \mu \mathbf{r}_{ij} \cdot \nabla \omega_{ij}}{(\rho_i + \rho_j)\,(r_{ij}^2 + \eta^2)} \mathbf{u}_{ij} + \mathbf{F}. \tag{3.1}$$

Similar expressions can be obtained for the operator defined in [99]. The pressure gradient is not considered here, while it can be included for $\mathbf{u}^*$ calculation, and the pressure calculation for time $n+1$ P^{n+1} from the Poisson solver will be the correction to P^n [17].

3.2.2 Pressure Poisson equation

The Laplacian and velocity divergence operators will be discretized by Eq. 2.20 and Eq. 2.15 respectively. The pressure Poisson equation can be written as

$$\sum_j 2\frac{V_j}{\rho_j}\frac{\left(P_i^{n+1}-P_j^{n+1}\right)\mathbf{r}_{ij}\cdot\nabla\omega_{ij}}{(r_{ij}^2+\eta^2)} = \frac{1}{\triangle t}\sum_j V_j\left(\mathbf{u}_j^*-\mathbf{u}_j^*\right)\cdot\nabla\omega_{ij}. \tag{3.2}$$

Therefore, a linear system $\mathbf{AX}=\mathbf{B}$ is generated, with

$$A(i,j) = -2\frac{V_j}{\rho_j}\frac{\mathbf{r}_{ij}\cdot\nabla\omega_{ij}}{(r_{ij}^2+\eta^2)}; \tag{3.3}$$

$$A(i,i) = -\sum_j A(i,j); \tag{3.4}$$

$$X(i) = P(i); \tag{3.5}$$

$$B(i) = \frac{1}{\triangle t}\sum_j V_j\left(\mathbf{u}_j^*-\mathbf{u}_j^*\right)\cdot\nabla\omega_{ij}. \tag{3.6}$$

It will be solved implicitly by Bi-CGSTAB solver [111].

3.2.3 Velocity at time $n+1$ $\mathbf{u}^{n+1}$

The velocity at the new time $n+1$ $\mathbf{u}^{n+1}$ is updated as

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\triangle t}{\rho} \sum_j V_j \left(P_j^{n+1} - P_i^{n+1}\right) \cdot \nabla \omega_{ij}. \tag{3.7}$$

And at the end the particle position at time $n+1$ $\mathbf{r}^{n+1}$ is updated following Eq. 2.29.

3.3 Linear solvers

Resolution of the linear systems is widely studied by mathematicians as the demand for an efficient and smoothly-converging solver increases from numerical simulations. In CFD, an efficient and robust solver also influences the application of the new method and code. Several solvers are widely used in academic and commercial codes: Gauss-Seidel, Conjugate Gradient (CG), Bi-Conjugate Gradient (Bi-CG), etc. Here two solvers, CG and Stabilized Bi-Conjugate Gradient (Bi-CGSTAB), applied in the code are introduced. It has been reported in several publications [37, 111] that the abnormal behavior of the residual norm in CG algorithm hinders the convergence, and by smoothing the residual, the convergence is highly enhanced, even monotonically, which is also observed in simulations, as shown in Fig. 3.2. In Fig. 3.2, for a 1-D SPH problem $\triangle \varphi = 0$ with $\frac{d\varphi}{dx} = 0$ at $x = 0$ and $\varphi = 0$ at $x = 1$, the convergence rates of linear solvers are plotted. The normalized residual, defined in Appendix B, is used as the convergence criterion set as 10^{-5}. The normalized residual is reduced monotonically and rapidly with the preconditioned

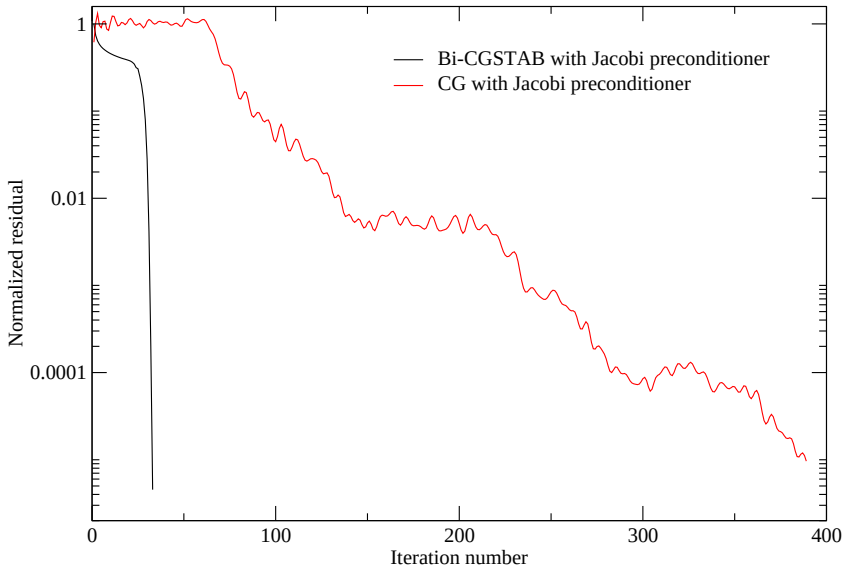


Figure 3.2: Convergence rate comparison between CG and Bi-CGSTAB linear solvers. A 1-D SPH problem $\triangle\varphi = 0$ with $\frac{d\varphi}{dx} = 0$ at $x = 0$ and $\varphi = 0$ at $x = 1$ has been solved to test the convergence speed of the linear solvers.

Bi-CGSTAB solver, while it converges irregularly with the preconditioned CG solver. Without special declaration, the same convergence criterion is used for all the simulations. The details about CG and Bi-CG solvers can be found in the Appendix B. In this thesis, the Bi-CGSTAB linear solver with the Jacobi Preconditioner is used in all the simulations.

One noteworthy point in ISPH method is that the convergence of pressure-velocity coupling solvers only depends on the solution of the linear system because the nonlinear convection term does not exist in Navier-Stokes (N.S.) equations in Lagrangian form, while the solution of the linear system only means the convergence of the internal iteration for segregated solvers in Finite Volume method (FVM) [29], and the convergence criterion for linear solvers in FVM is normally set as default value, such as 10^{-9} in Code Saturne [1]. Due to nonlinear convection terms in N-S equations, several external iterations are conducted for segregated solvers in FVM, and their convergence

can be decided by the relative difference of results from two consecutive external iterations [29].

3.4 Time-step constraints

In Weakly Compressible SPH (WCSPH) method, the choice of the time step is based on three limitations:

- Courant–Friedrichs–Lewy (CFL) condition
- Condition on the viscous diffusion
- Condition on the force per unit mass

All these three time-step constraints were adopted by Lee et al. in the Truly incompressible SPH method [63]. However only the first two conditions are considered to determine the time step in the most of ISPH literature as the third condition is actually limiting the distance between particles [77, 46] which can be avoided by the shifting of the particle position shown in §2.2.4 or simply overcome by the algorithm, such as ISPH_DI [100] and ISPH_DFDI [44]. Even with ISPH_DF [17] algorithm, the particle clustering does not depend on the time step, which has been tested in simulations of Taylor-Green vortices. In [17], Cummins and Rudman stated that the time-step constraint is different depending on the resolutions and the viscosity. Also, it is presented in [100] that the viscous diffusion constraint is used as a stability condition in the explicit finite difference method simulating viscous flows, and for simulations with high resolution or large viscosity, the diffusion constraint is more stringent than the CFL condition. In [44], in addition

to the CFL and diffusion condition, Hu and Adams were also considering the surface tension condition. In this work, both the CFL and diffusion conditions are considered as time step criteria.

CFL condition

The CFL condition is written as

$$\triangle t_{CFL} \leq \sigma \frac{h}{u_{ref}} \tag{3.8}$$

where, σ is a constant, and here it is set as 0.2; h is the smoothing length, which is proportional to the initial particle spacing δr; u_{ref} can be the numerical speed of sound c_0 for WCSPH and the maximum fluid velocity u_{max} for ISPH.

Viscous diffusion condition

The viscous diffusion condition is

$$\triangle t_{\nu} \leq \vartheta \, min \left(\frac{h^2}{\nu_{E,i}} \right) \tag{3.9}$$

where, ϑ is a constant, and here it is set as 0.2; $\nu_{E,i}$ is the effective viscosity for the particle i, and the minimum value for $\frac{h^2}{\nu_{E,i}}$ is taken into account. The effective viscosity $\nu_{E,i}$ is the kinematic viscosity ν in this work, as all the flows are considered as laminar flows.

3.5 Wall boundary tests

It has been introduced in §2.1.3 that the wall boundaries can be achieved through the mirror particle or dummy particle method. With the mirror particle method, the Neumann boundary can be easily built up through algebraic operations in the matrix. If particle j_m is the corresponding mirror particle of the fluid particle j as shown in Fig. 3.3, the discretized Poisson equation for particle i is

$$... + P(j) \cdot a(i,j) + ... + P(i) \cdot a(i,i) + ... + P(j_m) \cdot a(i,j_m) + ... = b(i). \quad (3.10)$$

For mirror particle j_m, there is the relation

$$P(j_m) = P(j). \quad (3.11)$$

Substituting Eq.3.11 into Eq.3.10, we can obtain

$$... + P(j) \cdot [a(i,j) + a(i,j_m)] + ... + P(i) \cdot a(i,i) + ... = b(i). \quad (3.12)$$

A similar linear system can be also obtained for dummy particle boundary,

$$... + P(j) \cdot a(i,j) + ... + P(i) \cdot a(i,i) + ... + P(k) \cdot [a(i,k) + a(i,k_{d1}) + a(i,k_{d2})] = b(i). \quad (3.13)$$

Tests for linear solver and the wall boundary condition are conducted with the Laplacian operator in [77]. A 1-D channel with periodic boundaries in horizontal direction is set up. The Poisson equation $\frac{d^2P}{dy^2} = 1$ is solved with a homogeneous Neumann wall boundary, $\frac{dP}{dy} = 0$ at $y = 1$, and a Dirichlet boundary, $P = 0$ at $y = 0$. The analytical solution for this problem

is $P = 0.5y^2 - y$. During all the tests, the convergence tolerance is set to 1.0×10^{-7} for the normalized residual. The Neumann wall boundary is set up with both dummy particle and mirror particle methods. In Fig. 3.4 the Poisson equation is solved with the same Laplacian operator Eq. 2.20 in §2.1.2 in Chapter 2 with a vertical resolution of 41 particles. The mirror particle method provides more accurate predictions for the wall boundaries than the dummy particle method. The non-uniform distribution of particles in the practical simulation may however complicate the accuracy behavior of the mirror-particle boundary method. The convergence is tested with different vertical resolutions, 40, 50 and 100 particles, and compared with the analytical solution, shown in Fig. 3.5.

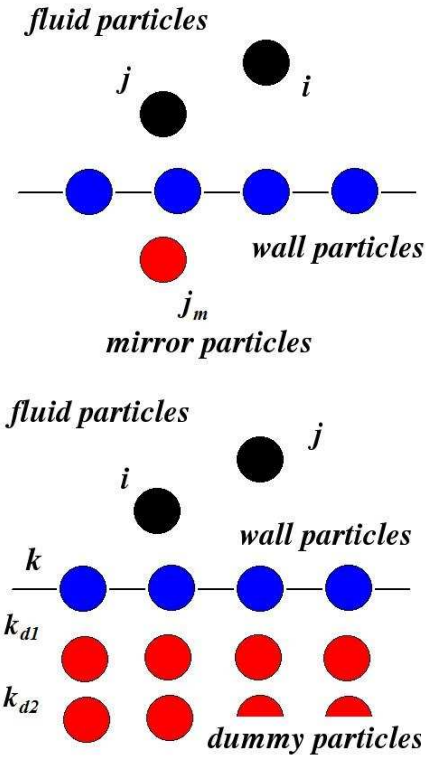


Figure 3.3: Mirror particle and dummy particle wall boundary. Black balls are fluid particles, blue wall particles, red mirror particles or dummy particles.

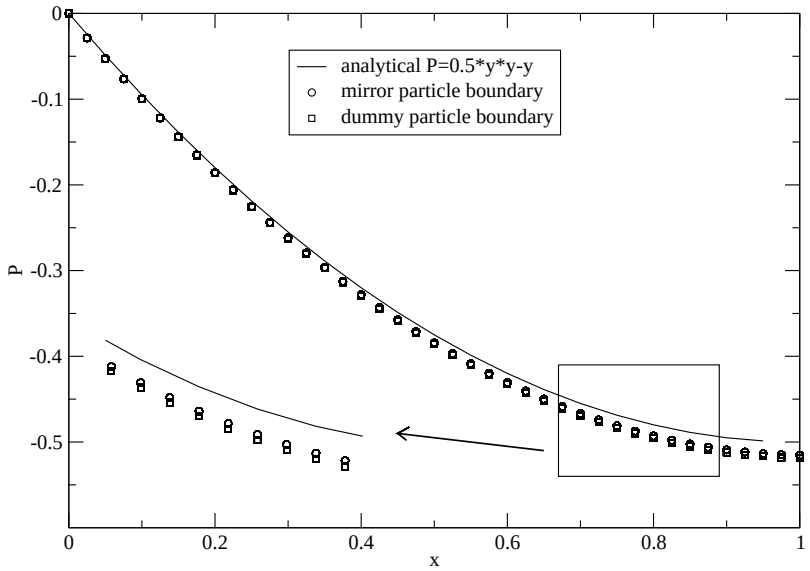


Figure 3.4: Boundary method investigation. A vertical resolution of 41 was used for both wall boundary methods.

3.6 Periodic boundaries

In this ISPH code, periodic boundaries have been implemented. Particles going out of the domain from one end will enter into the upstream end. For polar cell P shown in Fig. 3.6, not only the neighbouring cells, N, S, W, NW and SW , but also cells on the upstream end, NE, E and SE, are considered in the linked list for periodic boundaries. Therefore for particles close to different boundary ends, the distances between them are calculated from subtracting their coordinate difference from the domain length. For example, the horizontal distance $\triangle x$ for particles, i and j, at the two ends respectively is calculated as $\triangle x = L - (x_i - x_j)$, where $L = x_{max} - x_{min}$, and x_{max} and x_{min} are respectively the domain maximum and minimum horizontal coordinates, as presented in Fig. 3.6.

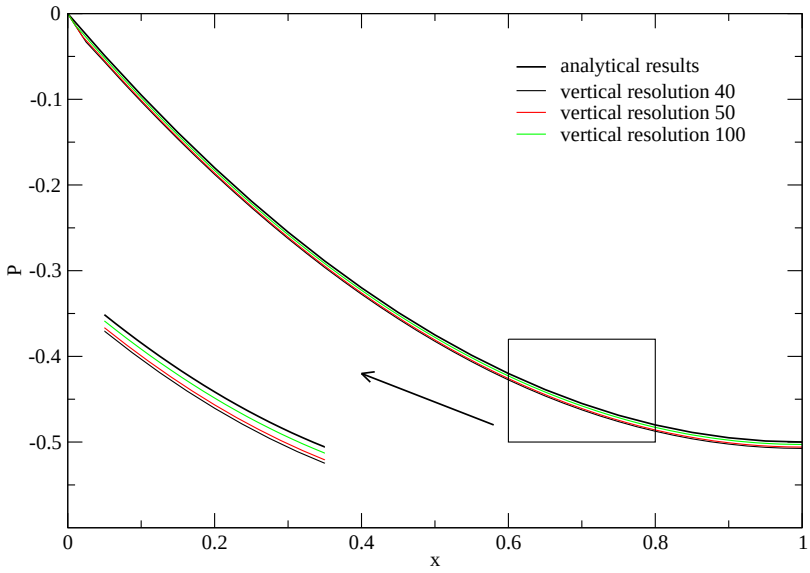


Figure 3.5: Linear solver and boundary conditions test.

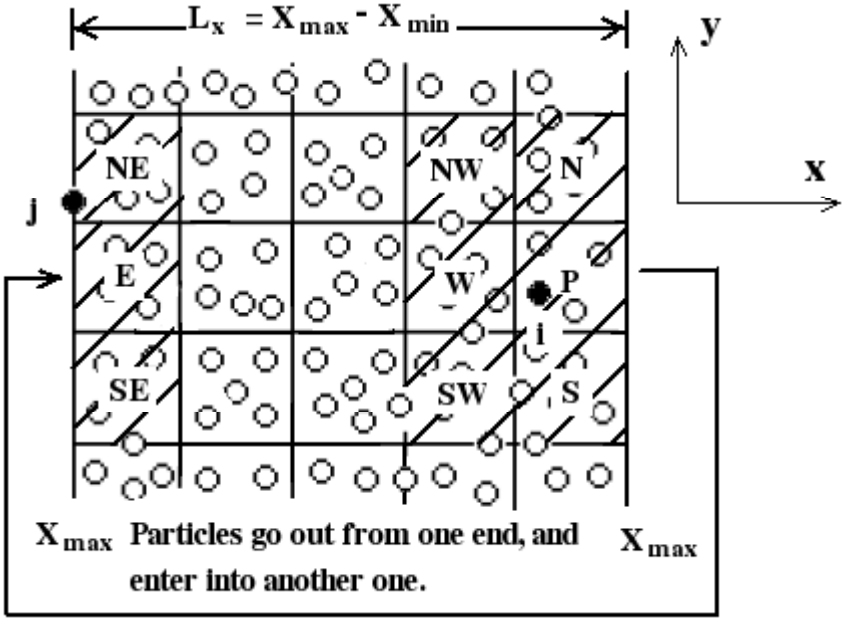


Figure 3.6: Illustration of periodic boundaries in the ISPH code.

3.7 Code structure

The code structure is detailed in Fig. 3.7. The geometry of the case is defined in a separate file, datclass.f. The geometric data is read into the solver. After choosing the ISPH solvers, ISPH_DI, ISPH_DFDI and ISPH_DFS, and the boundary methods, mirror particle and dummy particle boundary, the solver is ready for simulations. The ISPH_DF solver can be simply obtained by commenting out the particle shifting part in the ISPH_DFS solver. For the ISPH_DI, ISPH_DFDI and ISPH_DFS solvers, the corresponding viscous term and intermediate velocity/density will be calculated, and then the discretized Laplacian operator and the right hand side (R.H.S.), dependent on the solvers, are calculated, and the linear problem is solved. The velocity field is corrected with the pressure field obtained from the linear solver, and particle positions are updated. Differently, particle position are shifted and correction for hydrodynamic values are done in ISPH_DFS simulations. For ISPH_DFDI solver, an internal iteration for the converged particle position is conducted, a tolerance for the relative density change, defined as the R.H.S. in Eq.2.39, is normally 1% or 0.5%, and the maximum internal iteration normally is 20. The pressure and velocity field are corrected in a way similar to ISPH_DF.

3.8 Partial conclusion

Four incompressible SPH algorithms are applied with the help of open-source code SPHysics[121]. The choice of the solvers is performed by changing the switch in the code. Verlet's method [112] is used in the code to build up the

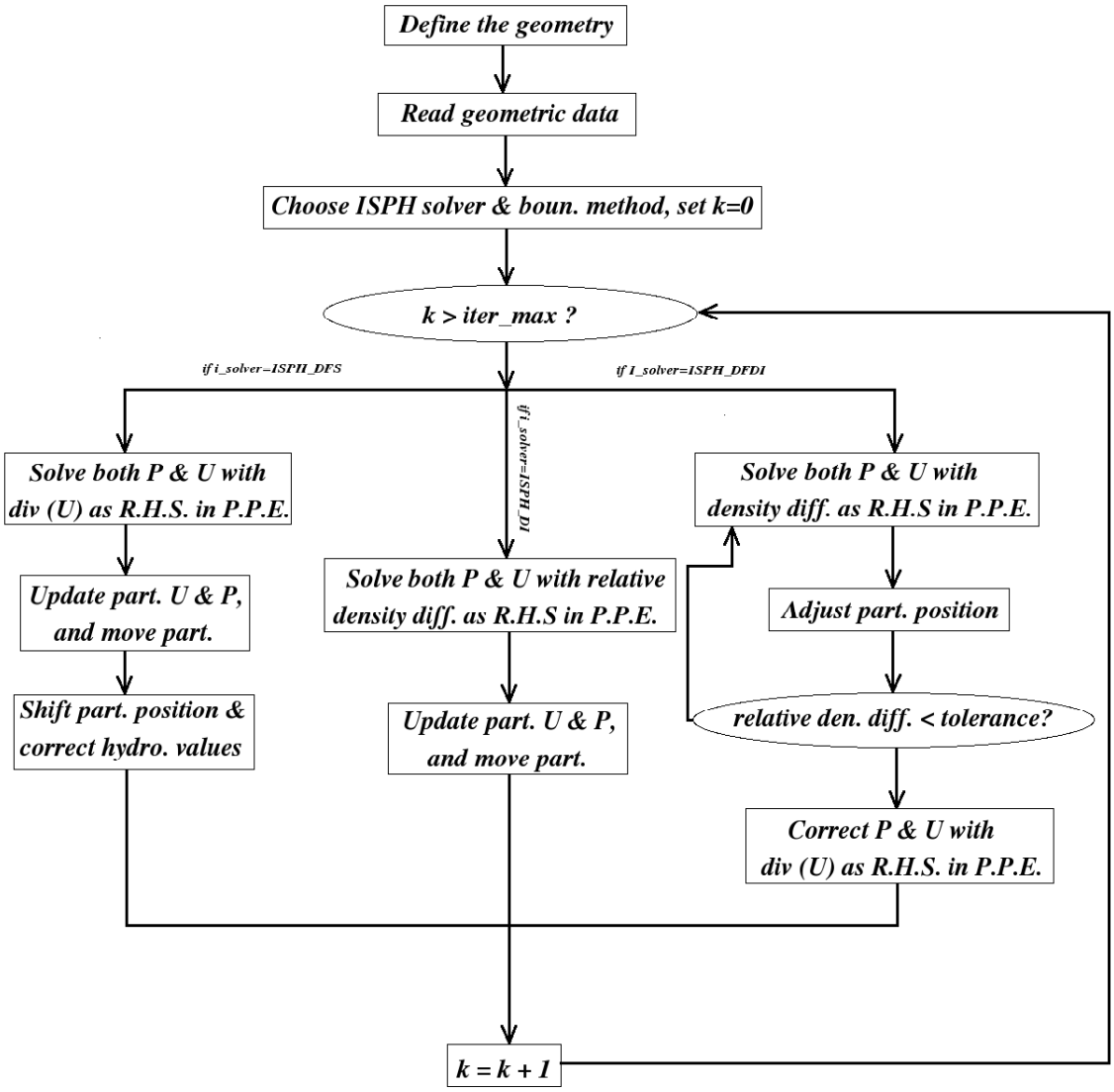


Figure 3.7: Code structure illustration. $iter\_max$ is the maximum iteration times; k is the iteration counter.

linked lists. A predictor-corrector scheme is applied. Two different linear solvers, preconditioned CG and preconditioned Bi-CGSTAB, are also available for simulations. It is observed that with preconditioned Bi-CGSTAB solver, the convergence of the linear problem is faster due to the residual smoothing procedure [37, 111]. The time step depends on the CFL condition and the viscous diffusion condition. Both dummy-particle and mirror-particle methods can be activated in the code to build up wall boundary condition. It has been shown that mirror-particle boundary method provides more accurate predictions for the wall boundaries than dummy particle method. The periodic boundaries are also available in the code.

Chapter 4

ISPH Method Comparisons and Accuracy Study for ISPH_DFS

As shown in the previous chapters, several incompressible SPH (ISPH) methods have been proposed in the past decade. This thesis is concerned only with the widely-used projection method for the pressure and velocity coupling. However, there is still no thorough analysis and comparison about the projection-based ISPH methods to the author's knowledge. The stability and accuracy of three methods which enforce either a divergence-free velocity field (ISPH_DF) [17, 63], density invariance (ISPH_DI) [100], or their combination (ISPH_DFDI) [44] are tested here through the standard Taylor-Green and spin-down vortex problems. In simulations with the ISPH_DF method, the truncation error may accumulate in certain situations, which leads to instability in simulations. This is numerically investigated in the following. The ISPH_DI method overcomes this problem by using the relative density difference as the right hand side (R.H.S.) in the pressure Poisson equation. But this method is inaccurate and noisy. The

ISPH_DFDI, combining advantages in ISPH_DF and ISPH_DI, separates the full time step into two half time steps: 1) Within the first half time step, the constant particle density is kept by an internal iteration, where the particles are shifted in a way similar to that in ISPH_DI method; 2) During the second half time step, the divergence-free velocity field is obtained by solving a pressure Poisson equation with the divergence of the intermediate velocity field as the R.H.S. This method can maintain accuracy and stability but at a high computational cost.

A new divergence-free ISPH approach, ISPH_DFS, is presented in Chapter 3. This method maintains accuracy and stability without increasing computational cost by slightly shifting particles away from streamlines while correcting their hydrodynamic characteristics, which can be observed from the discussion below. This should avoid the highly distorted particle spacing or particle clustering which causes instability, which is investigated here. The algorithm is not now analytically conservative, but importantly pressure fields are noise-free up to highest Reynolds numbers tested.

In this chapter, the accuracy and stability of the four projection-based ISPH methods are tested through two academic cases, Taylor-Green vortices and vortex spindown flows. Comparisons in terms of accuracy, stability and computing time are carried out.

4.1 Taylor_Green vortices

4.1.1 Case description

Simulations of Taylor-Green vortices are conducted. The analytical velocity field is

$$\begin{aligned} u &= -Ue^{\kappa t}cos(2\pi x)sin(2\pi y) \\ v &= Ue^{\kappa t}sin(2\pi x)cos(2\pi y) \end{aligned} . \tag{4.1}$$

And the pressure field is

$$P = -\frac{1}{4}e^{2\kappa t}\left[cos(4\pi x) + cos(4\pi y)\right]. \tag{4.2}$$

U is the velocity scale, equal to $1.0\,\mathrm{m/s}$ here; kinematic viscosity ν are $0.1\,\mathrm{m^2/s}$, $0.01\,\mathrm{m^2/s}$ and $0.001\,\mathrm{m^2/s}$ in three runs with three different Reynolds numbers, $Re = 10$, $Re = 100$, $Re = 1000$; $\kappa = -\frac{8\pi^2}{Re}$ is the decay rate of the velocity field; u and v are the horizontal and vertical velocity components respectively. Different resolutions are used to investigate the spatial accuracy of ISPH_DFS, which will be shown in Chapter 4. The Reynolds number is calculated by

$$Re = \frac{UD}{\nu} \tag{4.3}$$

where, D is the length of the unit square side. Because of the existence of analytical results for Taylor-Green vortices, the accuracy of the algorithm and the code can be tested here.

During simulations, it is observed that, under high Reynolds number situations, the particle spacings are highly compressed in one direction, but stretched in the other roughly normal direction, with ISPH_DF, shown in

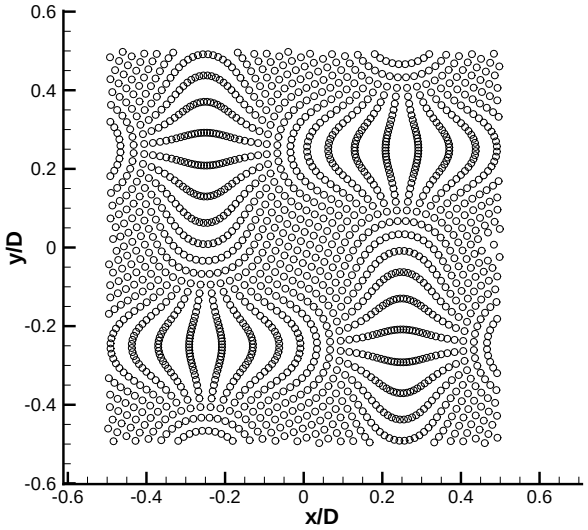


Figure 4.1: Highly distorted particle distributions in simulations of Taylor-Green vortices with ISPH_DF, with $Re = 1000$.

Fig.4.1. Because of the error caused by highly-distorted particle spacings, as mentioned before, the convergence with ISPH_DF may fail. For all values of Reynolds number, the four ISPH methods are applied. The accuracy, stability and computing expense of the four methods are compared.

4.1.2 Simulation results

Taylor-Green vortices are simulated in this work, with $Re = 10$, $Re = 100$, $Re = 1000$. For $Re = 10$, all four methods give stable and accurate solutions. However, when the Re number increases to 100, with ISPH_DF method the particle spacings are compressed in one direction, and stretched in the other direction roughly normal, as shown in Fig.4.1. When particles cluster together as happens at the four points $(\pm 0.25\ m, \pm 0.25\ m)$ in Fig.2.3, the error caused by the kernel flaw, presented in §2.1.4, will increase with decrease

in the particle spacing. The clustering cannot be avoided for high Reynolds number situations if particles move accurately along streamlines. It is observed that this error will accumulate, and jeopardize the simulation stability for the higher Reynolds number situations.

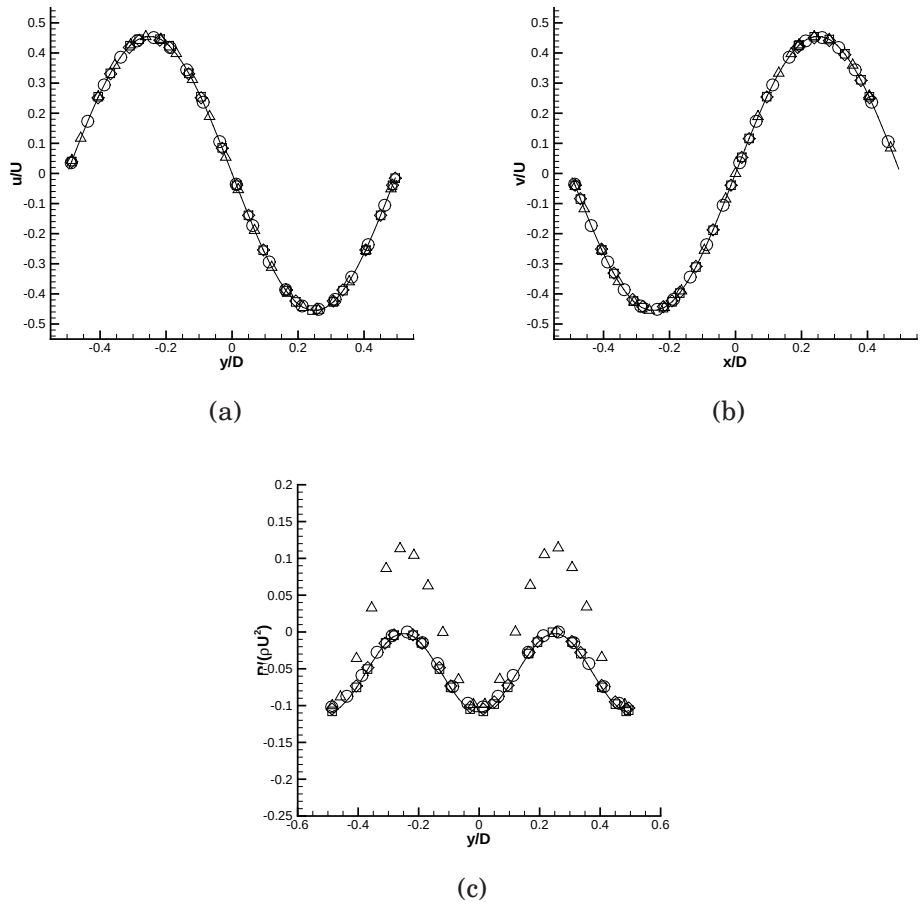


Figure 4.2: Normalized velocity and pressure profiles in Taylor-Green vortices, $Re = 10$, $t = 0.1s$, with a resolution of 40×40. Velocity components and distances are normalized by the velocity scale U, $U = 1$, and the square length D, $D = 1$. Pressure is normalized by ρU^2, and ρ is the fluid density. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$; (c) Normalised pressure profile, at $x/D = 0.0m$.
Δ= ISPH_DFDI; $\bigcirc$ = ISPH_DF; $\square$ = ISPH_DI; $\diamondsuit$= ISPH_DFS; — = Analytical results.

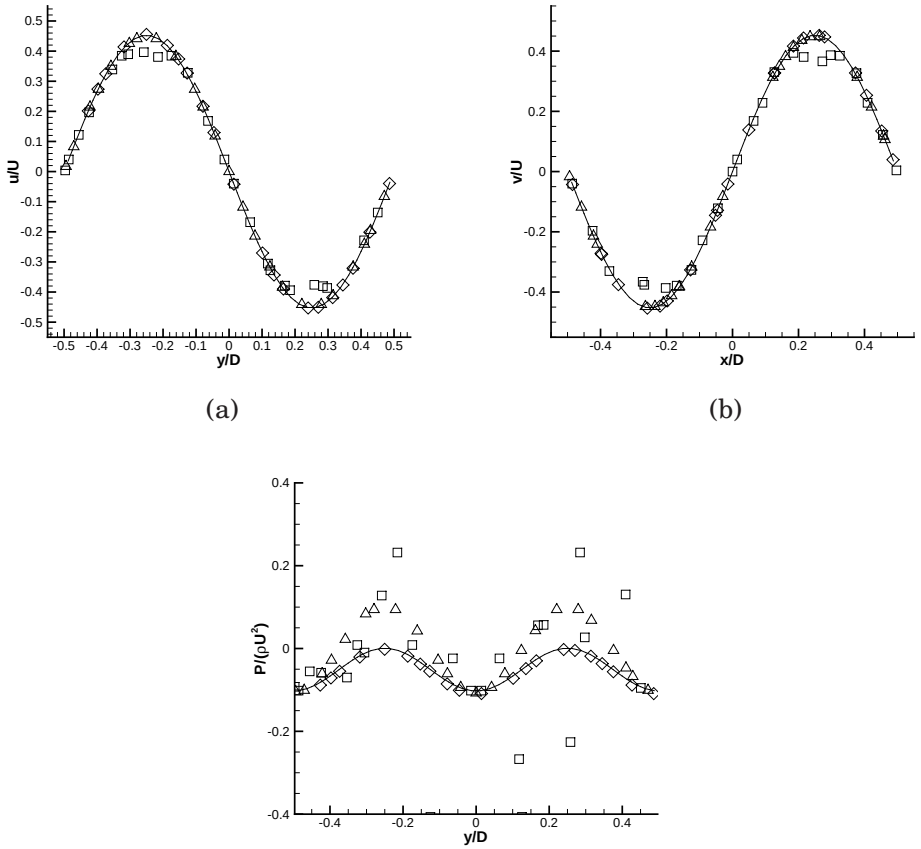
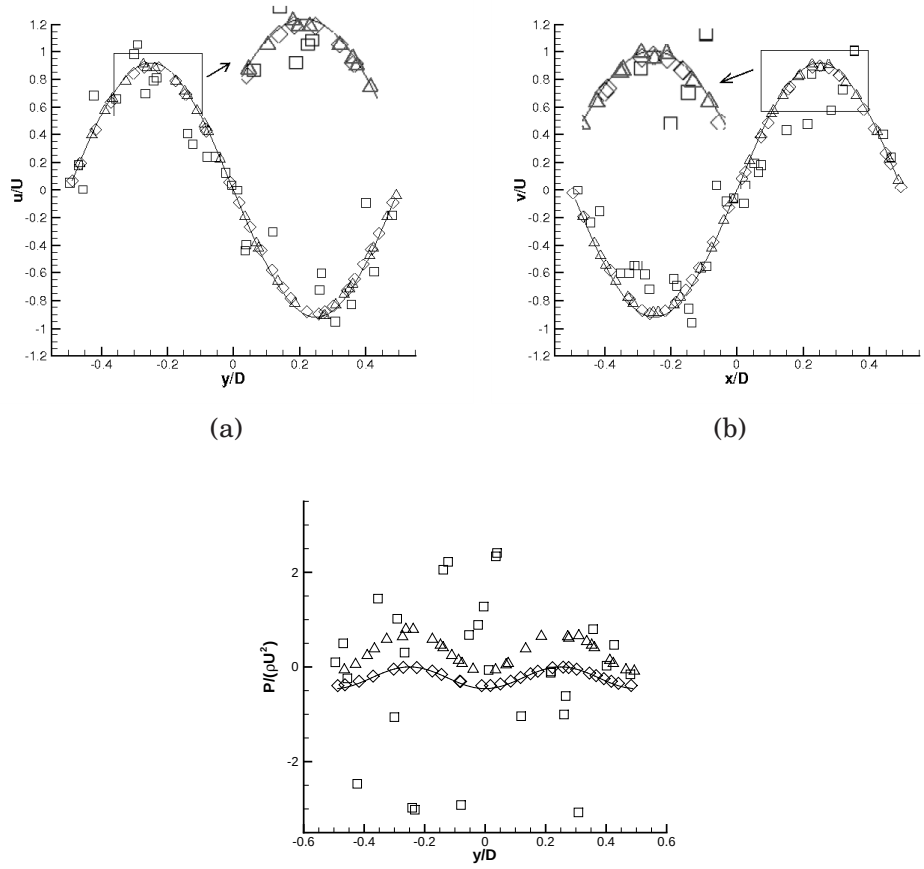


Figure 4.3: Normalized velocity and pressure profiles in Taylor-Green vortices, $Re = 100$, $t = 1.0s$, with a resolution of 40×40. Velocity components and distances are normalized with the velocity scale U, $U = 1$, and the square length D, $D = 1$. Pressure is normalized by ρU^2, and ρ is the fluid density. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$; (c) Normalised pressure profile, at $x/D = 0.0m$.
$\triangle$= ISPH_DFDI; $\square$ = ISPH_DI; $\diamond$= ISPH_DFS; — = Analytical results.



(a) (b)

(c)

Figure 4.4: Normalized velocity and pressure profiles in Taylor-Green vortices, $Re = 1000$, $t = 1.0s$, with a resolution of 40×40. Velocity components and distances are normalized with the velocity scale U, $U = 1$, and the square length D, $D = 1$. Pressure is normalized by ρU^2, and ρ is the fluid density. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$; (c) Normalised pressure profile, at $x/D = 0.0m$.
$\triangle$= ISPH_DFDI; $\square$ = ISPH_DI; $\diamondsuit$= ISPH_DFS; — = Analytical results.

Fig.4.2 presents the normalized velocity component, u, v and P profiles, at $x = 0.0\ m$ and $y = 0.0\ m$, $t = 0.1\ s$, with $Re = 10$; Fig.4.3 and 4.4 at time $t = 1.0\ s$, for $Re = 100$ and $Re = 1000$ cases respectively. Due to the stability

reason, for cases with high Reynolds numbers $Re = 100$ and $Re = 1000$ results from ISPH_DF method are not obtainable and not plotted. It can be observed in Fig.4.2 that all four projection-based ISPH methods can provide very good prediction for both velocity and pressure fields with $Re = 10$. When the Reynolds number increases to 100, ISPH_DF could not stably simulate the flow development, while the other three methods could continue simulations. In Fig.4.3, it is shown that ISPH_DFDI and ISPH_DFS predict both velocity and pressure field accurately; ISPH_DI underpredicts the velocities at $x = \pm 0.25\ m$ and $y = \pm 0.25\ m$. Increasing Reynolds number to 1000, ISPH_DFS provides smooth velocity and pressure profiles, but slightly underpredicts the velocity magnitudes; with ISPH_DI method, substantial numerical noise is generated; with ISPH_DFDI, some small numerical noise occurs at $y = \pm 0.25\ m$ and $x = \pm 0.25\ m$, shown in enlarged parts of velocity profiles in Fig.4.4. With ISPH_DFDI algorithm, all the pressure gradient fields are doubled as explained in §2.3.3.

Fig.4.5, 4.6 and 4.7 present the maximum velocity magnitude, u_{max}, decaying against time for $Re = 10$, $Re = 100$ and $Re = 1000$ respectively. For the low Reynolds case, $Re = 10$, all the methods stably simulate the flow development, and ISPH_DFS gives the most accurate prediction, shown in Fig.4.5. Increasing Reynolds number to, or over 100, only ISPH_DI, ISPH_DFDI and ISPH_DFS are presented, due to stability reasons. From Fig.4.6 and 4.7, it can be seen that ISPH_DI does not provide accurate prediction. ISPH_DFDI could simulate the flow with $Re = 100$. But with $Re = 1000$, and the same particle resolution, 40×40, certain numerical noise appears on the profile at the beginning of the flow development, shown in the enlarged part in Fig.4.7. Results with $Re = 1000$ were also not reported in [44]. Because ISPH_DFS

underpredicts the maximum velocity magnitude, u_{max}, with a resolution of 40×40, a run with a higher resolution of 80×80 is conducted. Convergence towards analytical results is obtained.

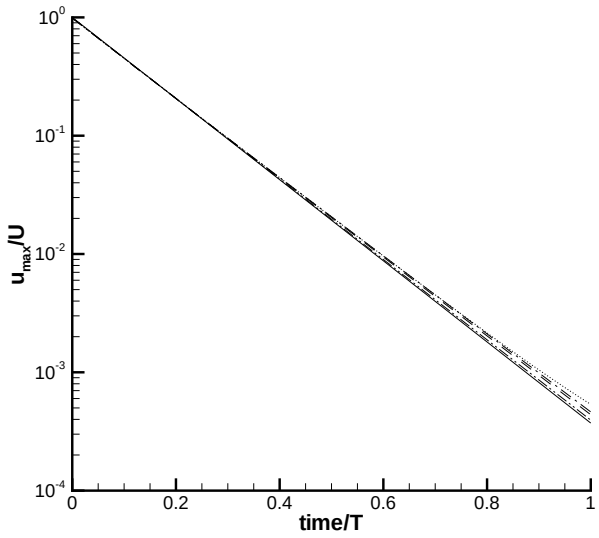


Figure 4.5: Maximum velocity, u_{max}, decaying against time in Taylor-Green vortices, $Re = 10$, with a resolution of 40×40. Maximum velocity, u_{max}, is normalized with velocity scale U; time is scaled by T, where $T = D/U$.
- - - = ISPH_DF; -·- = ISPH_DI; · · · = ISPH_DFDI; -···- = ISPH_DFS; — = Analytical results.

In Table 4.1 , the computing costs for the four methods are listed. ISPH_DFDI is the most time-consuming method. The computing cost of ISPH_DFS method is only slightly more than ISPH_DF and ISPH_DI.

All the four methods could manage to simulate the flow development with a low Reynolds number, $Re = 10$. ISPH_DF cannot stably simulate higher Reynolds cases. Although ISPH_DI could keep well distributed particle spacings, the numerical noise generated by this method contaminates the results under higher Reynolds number situations. ISPH_DFDI stably and accurately simulates cases with Reynolds numbers, $Re = 10$ and $Re = 100$. But

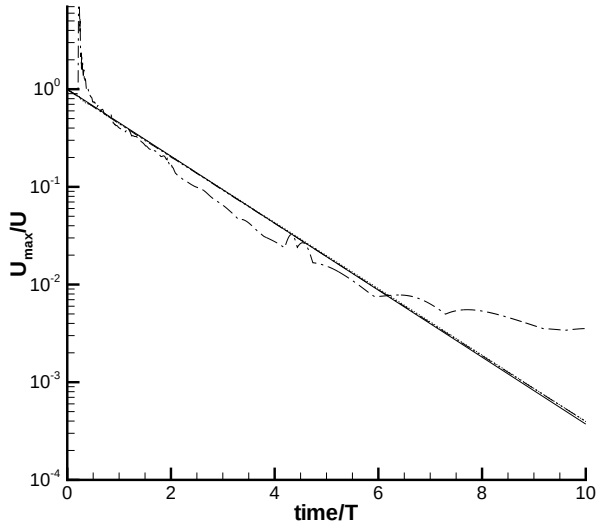


Figure 4.6: Maximum velocity, u_{max}, decaying against time in Taylor-Green vortices, $Re = 100$, with a resolution of 40×40. Maximum velocity, u_{max}, is normalized with velocity scale U; time is scaled by T, where $T = D/U$.
-·- = ISPH_DI; · · · = ISPH_DFDI; -···- = ISPH_DFS; — = Analytical results.

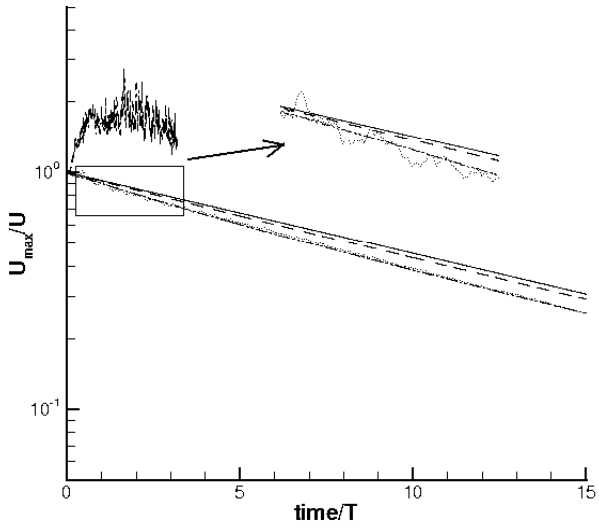


Figure 4.7: Maximum velocity, u_{max}, decaying against time in Taylor-Green vortices, $Re = 1000$. Maximum velocity, u_{max}, is normalized with velocity scale U; time is scaled by T, where $T = D/U$.
-·- = ISPH_DI(40×40); · · · = ISPH_DFDI(40×40); -···- = ISPH_DFS (40×40);
- - - = ISPH_DFS (80×80); — = Analytical results.

	$Re = 10$	$Re = 100$	$Re = 1000$
ISPH_DF	1,103 s	unstable	unstable
ISPH_DI	1,091 s	1,250 s	1,323 s
ISPH_DFDI	4,000 s	5,352 s	5,625 s
ISPH_DFS	1,213 s	1,281 s	1,387 s

Table 4.1: Computing costs comparison for Taylor-Green vortices with a resolution of 40×40. Physical time = 2.0 s. The same time step is used for all the methods for a given Reynolds number.

with $Re = 1000$, small numerical noise appears on the profile, shown in Fig. 4.7. ISPH_DFS efficiently stabilizes all simulations, and supplies predictions with less numerical noise for a given resolution.

4.2 Vortex Spin-down

4.2.1 Case description

Similar to Taylor Green vortices case, in vortex spin-down simulations, a vortex is bounded by four walls, placed in the middle of the domain, shown in Fig.4.8. The vortex spin-down cases, with $Re = 10$, $Re = 100$ and $Re = 1000$, are simulated. An initial velocity field is given by

$$\begin{aligned} u &= U(y - 0.5) \\ v &= U(0.5 - x) \end{aligned} \tag{4.4}$$

in a unit square. 41×41 particles, equivalent to a resolution of 40×40 in the Finite Volume (F.V.) method, are used to compare four projection-based

ISPH methods. The Reynolds number calculation is defined in the same way as for the case Taylor-Green vortices, Eq.4.3. Also, D, U and ν are the same as those in Taylor-Green vortices.

The accuracy, stability and computing time are compared among all the four ISPH methods. During simulations, it is found that particle spacings cannot be well maintained if ISPH_DF is used. It is the same as the findings in [27] that the irregular particle distribution hinders the convergence of the linear solver, and sometimes even causes the convergence to fail.

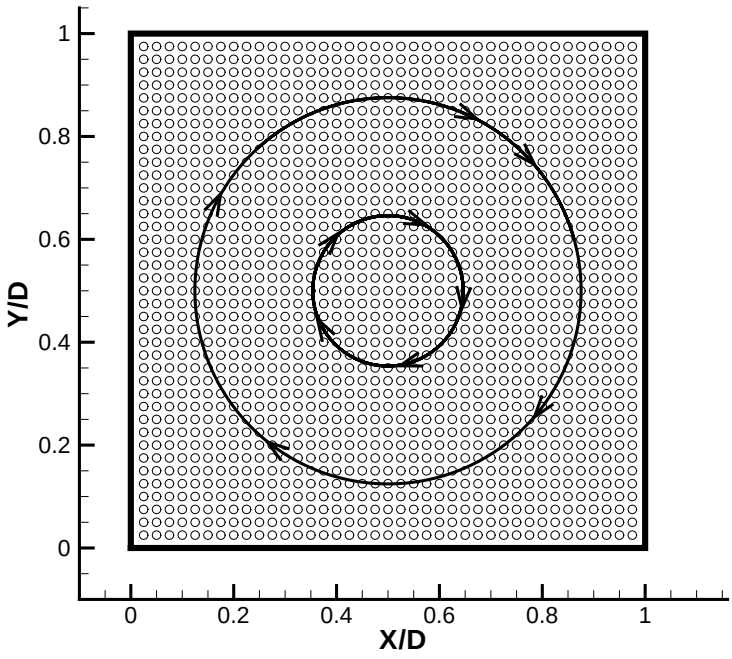


Figure 4.8: The geometry and initial velocity field of vortex spin-down case. Black solid line: wall; black lines with arrow: streamlines; circles: fluid particles in ISPH simulations.

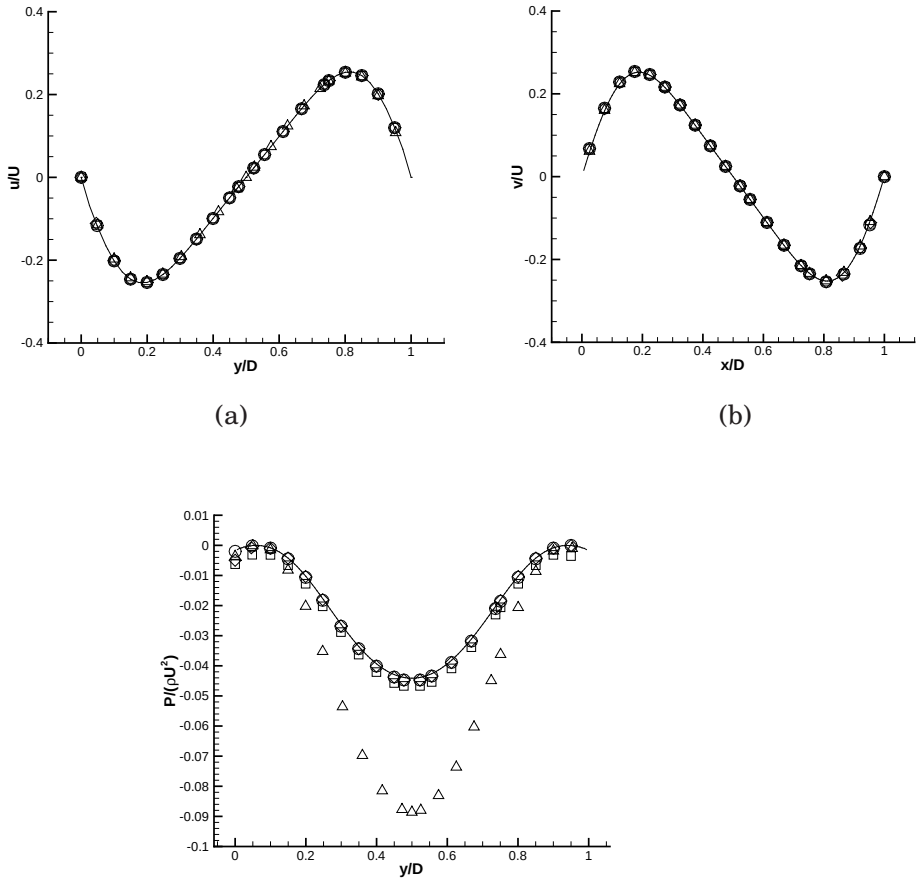
4.2.2 FV simulations for reference

STAR-CD is a finite-volume based commercial software. Because of the lack of analytical solutions and definitive data for the vortex spin-down cases, it is computed using STAR-CD producing numerically convergent solutions. The mesh convergence tests are considered for finite volume runs. The convergence criterion is set as 10^{-5}. The PISO algorithm is used for pressure and velocity coupling. Second-order differencing schemes are used for convection terms in the calculation.

4.2.3 Simulation results

In vortex spin-down simulations, cases with $Re = 10$, $Re = 100$ and $Re = 1000$ are simulated. Finite volume simulations, with STAR_CD commercial software, were run as comparison against projection-based ISPH methods. The mesh convergence test with uniform meshes, from 40×40, 80×80, to 160×160, is conducted in finite volume simulations. Converged results are obtained with a resolution of 160×160 and are used for the comparison. With $Re = 10$, all four methods manage to provide accurate predictions, shown in Fig.4.9. However, when the Reynolds number increases to 100, the simulation became unstable with ISPH_DF. As with the Taylor-Green vortex simulations, particle spacings are strongly distorted, which makes simulations quite unstable.

Fig.4.9 and 4.12 present the simulation results with $Re = 10$ situation using all four ISPH methods. All the methods provide very good prediction for the velocity decay. Note that for ISPH_DFDI pressure is used as an inter-



(a) (b)

(c)

Figure 4.9: Velocity and pressure profiles in vortex spin-down, $Re = 10$, $t = 0.1s$, with a resolution of 40×40. Velocity components are normalized with the velocity scale U, $U = 1$; the coordinates are normalized with the square length D, $D = 1$; the pressure is normalized with ρU^2, where ρ is the fluid density. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$; (c) Pressure profile at $x/D = 0.0m$. Δ = ISPH_DFDI; $\bigcirc$ = ISPH_DF; $\square$ = ISPH_DI; $\diamondsuit$ = ISPH_DFS; — = STAR-CD.

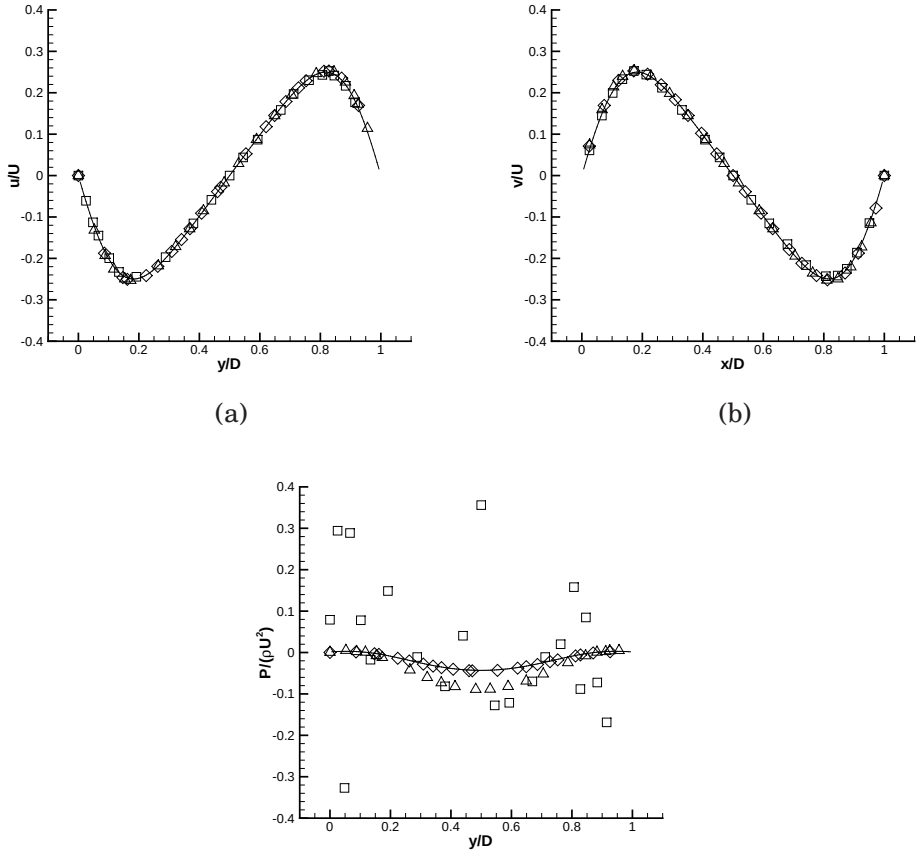


Figure 4.10: Velocity and pressure profiles in vortex spin-down, $Re = 100$, $t = 1.0s$, with a resolution of 40×40. Velocity components are normalized with the velocity scale U, $U = 1$; the coordinates are normalized with the square length D, $D = 1$; the pressure is normalized with ρU^2, where ρ is the fluid density. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$; (c) Pressure profile at $x/D = 0.0m$.
Δ = ISPH_DFDI; $\square$ = ISPH_DI; $\diamondsuit$ = ISPH_DFS; — = STAR-CD.

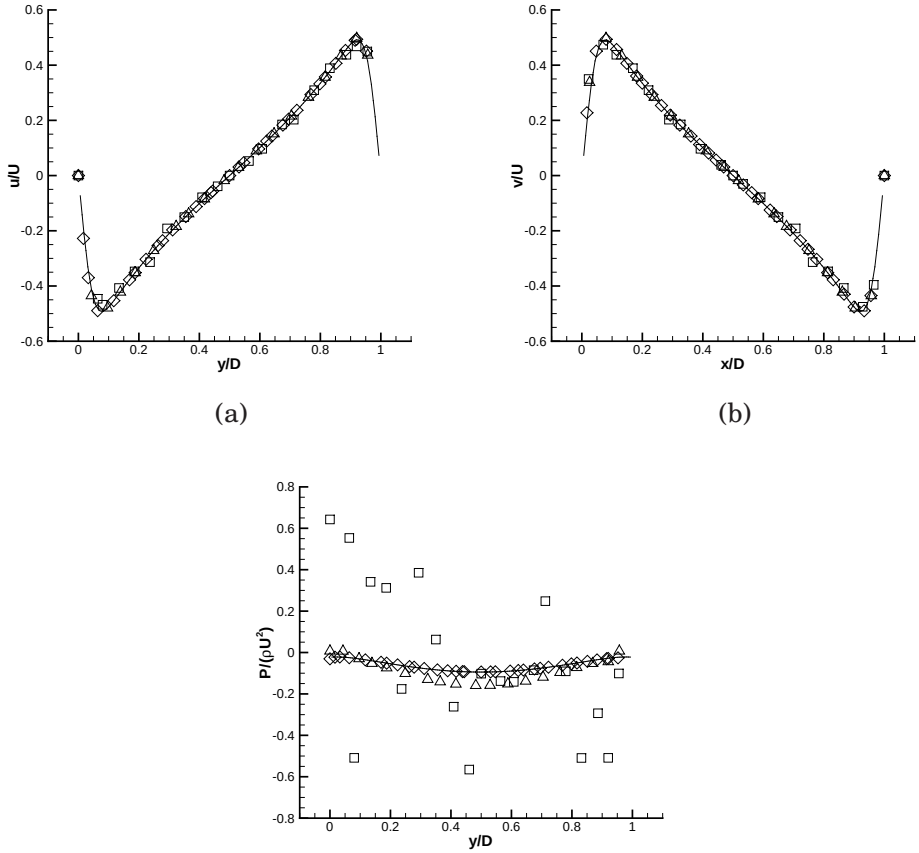


Figure 4.11: Velocity and pressure profiles in vortex spin-down, $Re = 1000$, $t = 1.0s$, with a resolution of 40×40. Velocity components are normalized with the velocity scale U, $U = 1$; the coordinates are normalized with the square length D, $D = 1$; the pressure is normalized with ρU^2, where ρ is the fluid density. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$; (c) Pressure profile at $x/D = 0.0m$.
Δ = ISPH_DFDI; $\square$ = ISPH_DI; $\diamondsuit$ = ISPH_DFS; — = STAR-CD.

	$Re = 10$	$Re = 100$	$Re = 1000$
ISPH_DF	913 s	unstable	unstable
ISPH_DI	934 s	997 s	1,233 s
ISPH_DFDI	1,262 s	5,339 s	7,679 s
ISPH_DFS	965 s	1,067 s	1,395s

Table 4.2: Computing costs comparison for vortex spin-down with a resolution of 40×40. Physical time $t = 2.0s$. The same time step is used for all the methods under the same Reynolds situation.

nal mechanism to drive velocity. Since this is applied in the correction at the second half time step, Eq.2.46, the computed pressure appears as twice the actual pressure. For $Re = 100$ or higher, ISPH_DF does not provide stable simulations, as explained before. With ISPH_DI, the simulation is stabilized. However, this method could not predict the fluid field accurately. Much numerical noise appears in the simulations, shown in Fig.4.11, 4.13 and 4.14. ISPH_DFDI and ISPH_DFS both accurately predict the flow development under the situation of $Re = 100$, presented in Fig.4.10 and 4.13. For $Re = 1000$, they both underpredict the velocity decaying rate although the velocity field is well predicted at a physical time of 1 s, shown in Fig.4.11 and 4.14. With $Re = 1000$, ISPH_DFDI produces small numerical noise at the beginning of flow development, shown in the enlarged part in Fig.4.14.

In Table 4.2, the computing costs of all four methods are listed. Increasing Reynolds number from 10 to 1000, the computing expense increases markedly with ISPH_DFDI. For higher Reynolds numbers, the internal iteration in ISPH_DFDI, Eq.2.44, sometimes needs more than 10 cycles to regularize particle spacing. It can be seen that ISPH_DFS is only slightly more computationally expensive than ISPH_DF and ISPH_DI, while providing better stability and accuracy.

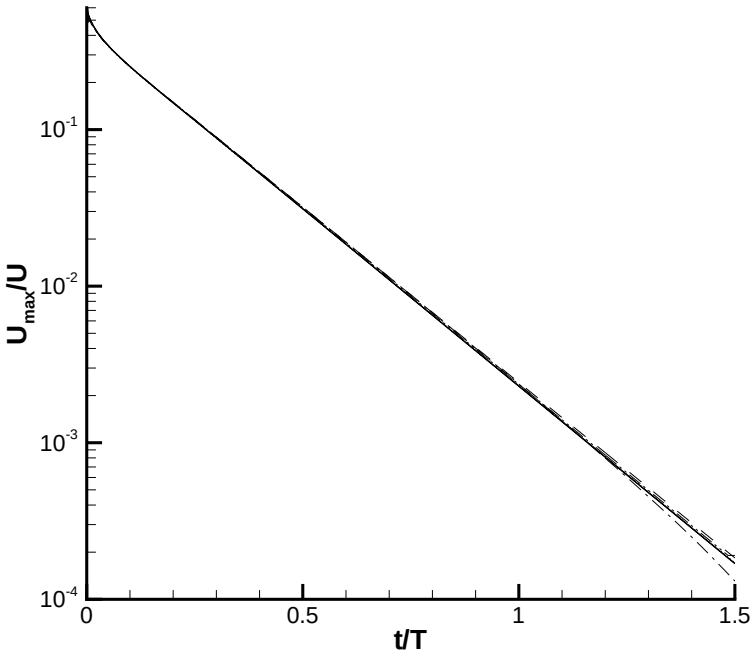


Figure 4.12: Maximum velocity magnitude, U_{max}, decaying against time in vortex spin-down, $Re = 10$, with a resolution of 40×40. Maximum velocity, U_{max}, is normalized with velocity scale U; the physical time, t, is normalized with T, where $T = U/D$. - - - = ISPH_DF; -·- = ISPH_DI; · · · = ISPH_DFDI; -···- = ISPH_DFS; — = STAR-CD.

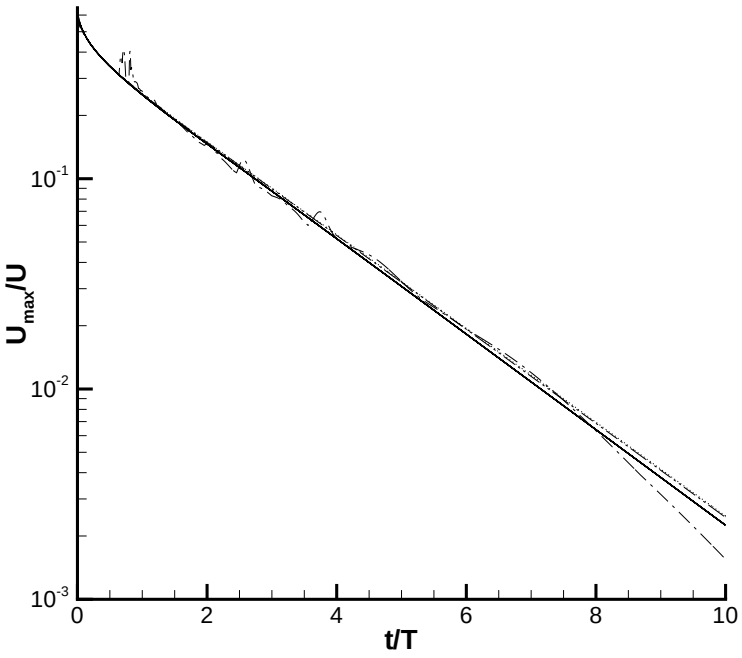


Figure 4.13: Maximum velocity magnitude, U_{max}, decaying against time in vortex spin-down, $Re = 100$, with a resolution of 40×40. Maximum velocity, U_{max}, is normalized with velocity scale U; the physical time, t, is normalized with T, where $T = U/D$. -·- = ISPH_DI; · · · = ISPH_DFDI; -···- = ISPH_DFS; — = STAR-CD.

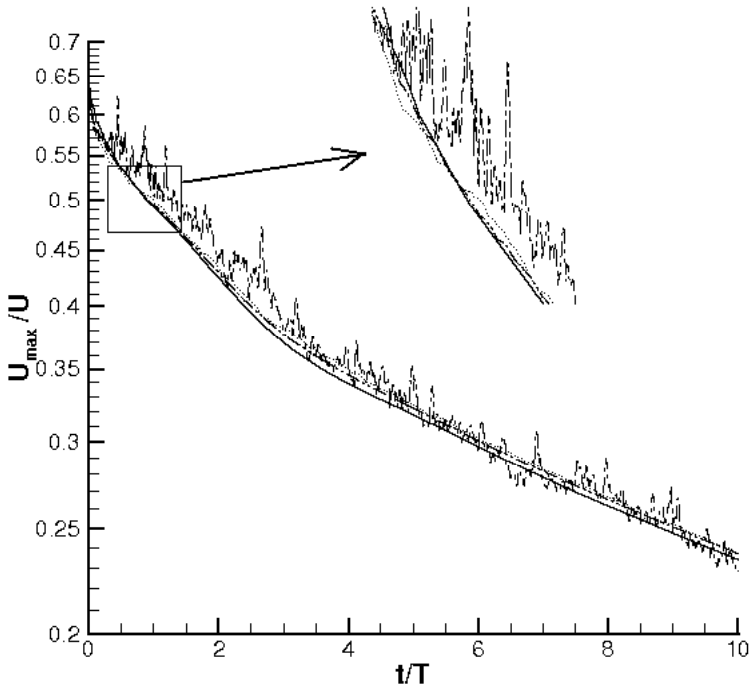


Figure 4.14: Maximum velocity magnitude, U_{max}, decaying against time in vortex spin-down, $Re = 1000$, with a resolution of 40×40. Maximum velocity, U_{max}, is normalized with velocity scale U; the physical time, t, is normalized with T, where $T = U/D$.
-·- = ISPH_DI; · · · = ISPH_DFDI; -···- = ISPH_DFS; — = STAR-CD.

4.3 Accuracy test with Taylor-Green vortices

4.3.1 Relative error definition

To quantify the error in the ISPH_DFS methods, profiles of an estimated relative error are plotted out. In [44, 9], a relative error, ε, is calculated, and defined as

$$\varepsilon = \mid \frac{\phi_{ex,max} - \phi_{ISPH,max}}{\phi_{ex,max}} \mid, \tag{4.5}$$

where, $\phi_{ex,max}$ and $\phi_{ISPH,max}$ are respectively the exact and simulated maximum values of a general variable, ϕ. To better investigate the error, an estimated relative error ε_{L2}, based on relative L2 norm, is also calculated, given by

$$\varepsilon_{L2} = \left(\frac{\sum_{i=1}^{n_p} \left(\phi_{ex,i} - \phi_{ISPH,i}\right)^2 v_i}{\sum_{i=1}^{n_p} \phi_{ex,i}^2 v_i} \right)^{\frac{1}{2}}, \tag{4.6}$$

where n_p is the number of particles; v_i is the volume for particle i, which is constant here; $\phi_{ex,i}$ and $\phi_{ISPH,i}$ are respectively the exact and simulated general variable value for particle i. Due to the symmetry of flow pattern in Taylor-Green vortices, only the error for horizontal velocity component, u, is calculated in relative L2 norm. The corresponding L2-norm based absolute

error is

$$e_{L2} = \left(\frac{\sum_{i=1}^{n_p} \left(\phi_{ex,i} - \phi_{ISPH,i}\right)^2 v_i}{\sum_{i=1}^{n_p} v_i} \right)^{\frac{1}{2}}, \tag{4.7}$$

As discussed in §2.1.2, SPH operators can achieve higher accuracy through algebraic reasoning of the Taylor expansion. In [10, 11] Chen et al. by inverting the whole matrix for each particle, normally 3×3 for 2-D problem and 6×6 for 3-D problem, calculate the whole Hessian ($\phi_{x_i x_j}$), and achieve 2nd-order accuracy. However, this demands a huge computing expense, for large simulations with millions of particles almost practically impossible. Schwaiger [99] suggested a new Laplacian operator with higher accuracy by including the gradient terms and higher-order terms in the Taylor expansion, as presented in §2.1.2. Here through the Taylor-Green vortex test case, the influence of viscous and Laplacian operators on the simulation accuracy is studied, and the difficulty in achieving high-accuracy simulations is discussed.

4.3.2 Accuracy and convergence rate of ISPH_DFS

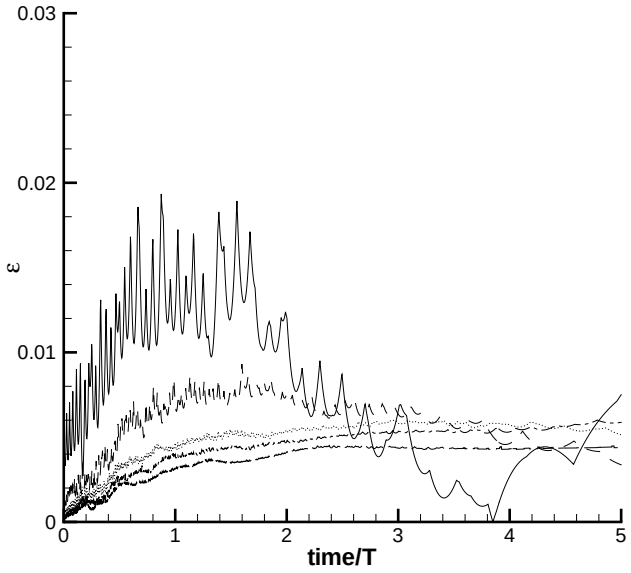
The viscous and Laplacian operators are divided into two groups here, the one presented in [77] called lower-order operators here and the one proposed by Schwaiger [99], named higher-order operators. Fig. 4.15 and 4.16 show time variations of estimated error with the viscous and Laplacian operators in [77] for $Re = 100$ and $Re = 1000$ respectively. Based on the relative L2 norm, seeing Eq. 4.6, two maximum relative error values, 0.045% and 2.5%, for the finest meshes are obtained respectively at $Re = 100$ and $Re = 1000$

for the lower-order operators. For $Re = 1000$, a spatial convergence speed is also estimated for both higher-order [99] and lower-order [77] operators, shown in Fig. 4.17. An estimated absolute L2-norm value, e_{L2}, is calculated, at $t = 2.0s$. The estimated spatial convergence speed is close to first order for ISPH_DFS with both two groups of viscous and Laplacian operators, with slightly more accurate prediction from higher-order operators. The same conclusion was also obtained in [63] for ISPH method. And in [110], the same first-order convergence speed is also obtained with the kernel correction for gradient and divergence operators [4, 82].

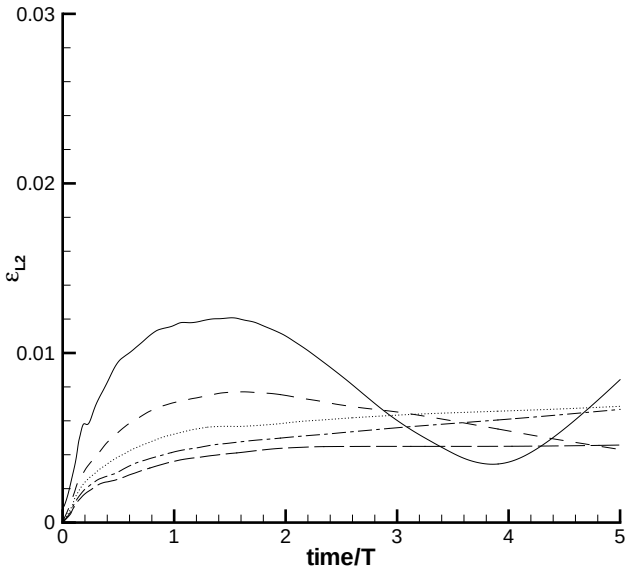
4.3.3 Influence of correction stage on the accuracy

By using Eq. 2.21 in §2.1.2, the accuracy of viscous and Laplacian operators is improved [99]. The convergence speed may also be possibly improved. However, with only first-order Taylor expansion in the correction stage, Eq. 2.49, the convergence speed is limited. To give an order consistent with or higher than the improved operators, a simple expansion for the second-order gradient operator, Eq. 2.15 with the normalised kernel Eq. 2.16, is applied here as

$$\nabla(\nabla\phi) \simeq \sum_j V_j \left(\sum_k V_k(\phi_k - \phi_j)\nabla W_{jk} - \sum_l V_l(\phi_l - \phi_i)\nabla W_{ij} \right) \nabla W_{ij}, \quad (4.8)$$

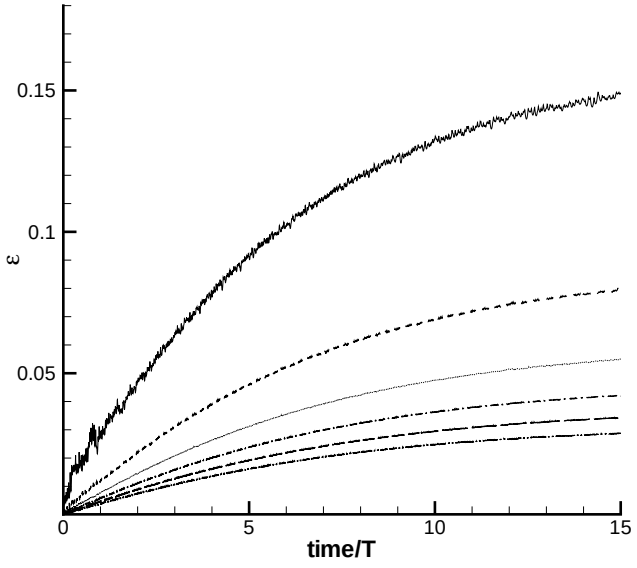


(a)

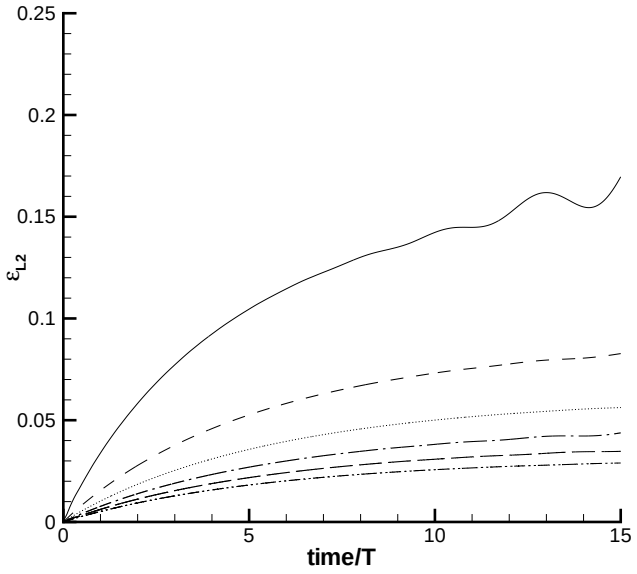


(b)

Figure 4.15: Estimated relative error profiles for the horizontal velocity component, u, in Taylor-Green vortices with different resolutions, $Re = 100$. (a) estimated error based on Eq.4.5; (b) estimated error based on L2 norm. The time is normalised by T, and $T = U/D$.
— = 40×40; - - - = 80×80; $\cdots$ = 120×120; -·- = 160×160; — — = 200×200.



(a)



(b)

Figure 4.16: Estimated relative error profiles for the horizontal velocity component, u, in Taylor-Green vortices with different resolutions, $Re = 1000$. (a) estimated error based on Eq.4.5; (b) estimated error based on L2 norm. The time is normalised by T, and $T = U/D$.
— = 40×40; - - - = 80×80; · · · = 120×120; -·- = 160×160; — — = 200×200; -···- = 240×240.

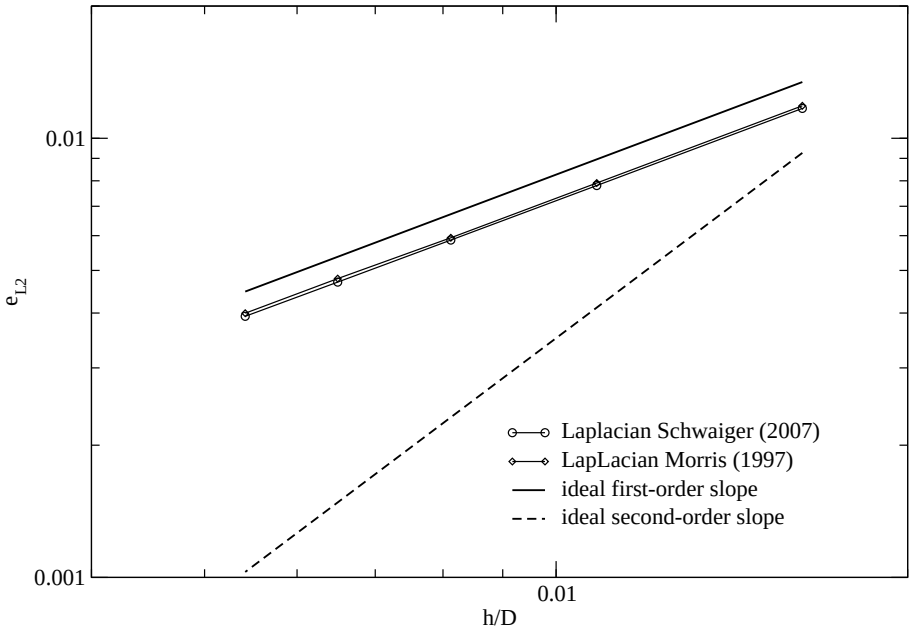


Figure 4.17: Convergence speed estimation for both Laplacian operators in [77] and in [99], based on the L2 error estimation of the horizontal velocity component, u, in Taylor-Green vortices with ISPH_DFS, $Re = 1000$, $t = 10.0\,s$. The smoothing length, h, is normalised by the square length D, where $D = 1$.

where, ∇W_{ij} and ∇W_{jk} are normalised kernels. By considering the whole Hessian, the Taylor expansion is here written as

$$\phi_{i'} = \phi_i + (\nabla\phi)_i \cdot \delta\mathbf{r}_{ii'} + \frac{1}{2}\delta\mathbf{r}_{ii'}\nabla(\nabla\phi)\delta\mathbf{r}_{ii'} + \mathcal{O}(\delta r_{ii'}^3). \tag{4.9}$$

In Fig.4.18, the L2-norm based errors are plotted for different correction situations, and both corrections, including the first derivative terms in Taylor expansion and including the second derivative terms in Taylor expansion, gave almost the same prediction, more accurate than the simulation without correction stage. Therefore, more terms in the correction stage do not increase the accuracy, producing more or less similar predictions. The error in the simulation without correction is still bounded. The reason why the bounded error is obtained even without the Taylor-expansion correction is possibly that the shifting of particle positions is more like “a random walk”; that is, an extra diffusion is introduced, which makes the algorithm still stable even without correction.

4.4 Partial conclusion

The accuracy and stability of three existing projection-based incompressible SPH (ISPH) methods have been tested for Taylor-Green and vortex spindown flows. A new projection-based ISPH method is introduced which maintains stability and accuracy without loss of efficiency. It can be concluded that:

1. ISPH_DF provides accurate predictions in certain cases. However, when

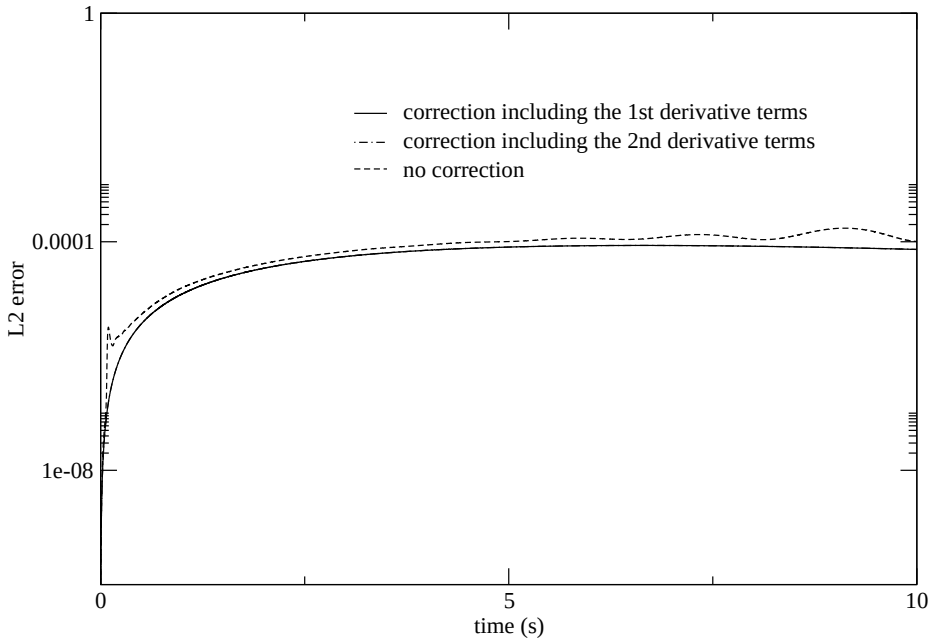


Figure 4.18: The influence of correction stage to the simulation accuracy in the Taylor-Green vortices test case. Simulations are conducted with a particle resolution 160×160.

the particle spacing becomes highly distorted, which occurs at high Reynolds number, errors caused by anisotropic particle spacings accumulate. ISPH_DF exhibits instability in such cases.

2. ISPH_DI overcomes the instability introduced by irregularly-distributed particles. However, it is shown that this method does not give accurate predictions for these flows. The numerical noise can be extremely high.

3. By combining the accuracy characteristics of ISPH_DF and the stability of ISPH_DI, ISPH_DFDI produces accurate and stable simulations, with evenly distributed particles. However, efficiency is sacrificed, as the method often requires many internal iterations, and it gives doubled pressure and pressure gradient fields. It can be observed in Fig.4.7 and Fig.4.14 that, with the same numerical parameters, ISPH_DFDI could introduce small numerical noise in the result.

4. Based on ISPH_DF, ISPH_DFS provides a stable algorithm without sacrificing efficiency. Particles are shifted slightly across streamlines, avoiding particle spacing distortion and the error resulting from particle clustering, and their hydrodynamic properties are adjusted by Taylor series interpolation.

5. The accuracy of the new approach ISPH_DFS is tested through Taylor-Green vortices. The new viscous and Laplacian operators, proposed by Schwaiger [99], are also applied, and compared with the operators in [77]. Both operators gave the first-order spatial convergence speed, even with higher accuracy from operators in [99]. The influence of the Taylor-expansion correction, with consideration of different terms in Taylor expansion, is also studied. Having second-derivative terms in the correction stage does not obviously improve the prediction accuracy. However, it shows that the correction of hydrodynamic values by Taylor expansion is necessary.

Chapter 5

Validation by Benchmark Test Cases

In this Chapter, the accuracy of ISPH_DFS method will be analyzed through two benchmark test cases. The lid-driven cavity and bluff body, normally squares or circular cylinders, are often used as benchmark test cases to assess new codes and algorithms. Here, the lid-driven cavity and the flow around a circular cylinder, at various Reynolds numbers, are simulated. In this Chapter, except where stated otherwise the ISPH method used is ISPH_DFS method introduced in Chapter 2.

5.1 Lid-driven cavity

5.1.1 Case description

The lid-driven cavity problem has long been used as a test or validation case for new codes or new solution methods. A good set of data for comparison is [32], where the data are listed out in a table for different Reynolds number situations. The two-dimensional geometry is shown in Fig.5.1. Although the lid-driven cavity case is often simulated as a steady case, in simulations, the fluid is accelerated by the lid at the top of the cavity to a steady state. Simulations with fully-developed flows are compared with numerical data in [32] and STAR_CD results.

In this test case, the Reynolds number is calculated by

$$Re = \frac{UD}{\nu}. \tag{5.1}$$

U and D are set to 1 m/s and 1 m for the convenience in the normalization and the calculation of Reynolds number. Three simulations with different Reynolds numbers, $Re = 100$, $Re = 400$ and $Re = 1000$, are simulated.

To obtain convergence towards the data set in [32], high resolutions are used in high Reynolds number situations. Tab.5.1 shows different resolutions used in different Reynolds situations. The time step is decided based on the analysis in §2.3.

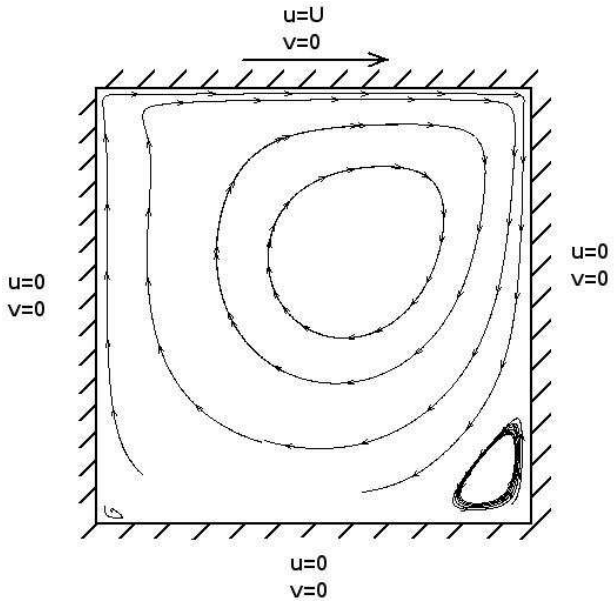


Figure 5.1: Geometry of lid driven cavity case. The graph shows the streamline in the case with $Re = 400$, simulated by ISPH_DFS, with a resolution of 61×61.

	$Re = 100$	$Re = 400$	$Re = 1000$
Resolution 1	41×41	81×81	81×81
Resolution 2	81×81	161×161	161×161
Resolution 3	161×161	241×241	241×241

Table 5.1: Resolutions used in different Reynolds situations

5.1.2 Simulations with STAR-CD

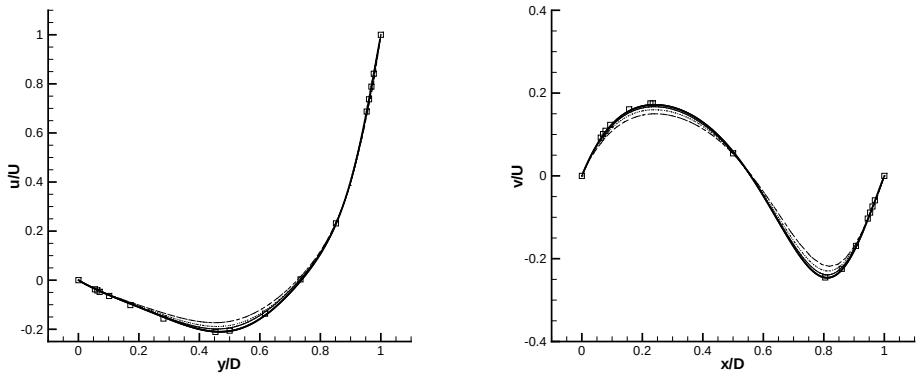
In [32], there is no reference data for pressure field. Therefore, the lid-driven cavity cases are simulated by STAR-CD as well to provide comparison objects. The mesh convergence tests are considered for finite volume runs. The convergence criterion is set as 10^{-5}. The PISO algorithm is used for pressure and velocity coupling. Second-order differencing schemes are used for convection terms in the calculation.

5.1.3 Simulation results and discussion

As a widely-studied test case, the lid driven cavity is used to further validate the ISPH_DFS algorithm. Reynolds numbers, $Re = 100$, $Re = 400$, and $Re = 1000$, are simulated and results are compared with data from [32]. Convergence tests for ISPH_DFS, with different resolutions are conducted for all Reynolds numbers.

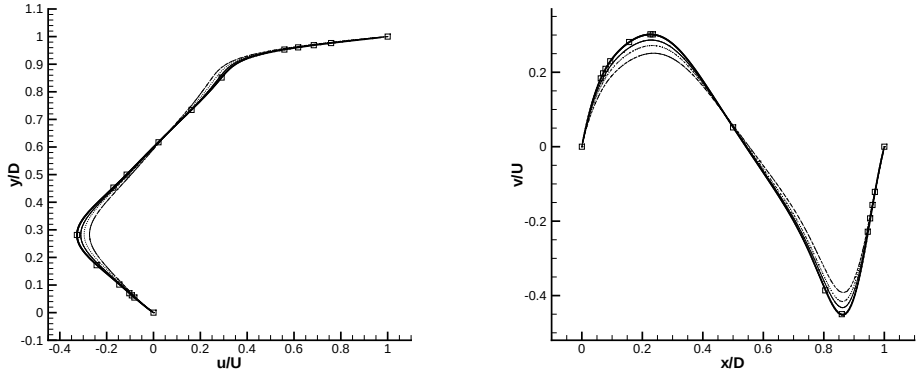
In Fig.5.2, 5.3 and 5.4, the velocity profiles in the middle of the domain, $x/D = 0.5\ m$ and $y/D = 0.5\ m$, are presented for each Reynolds number, $Re = 100$, $Re = 400$, $Re = 1000$, respectively. With increasing the resolution, the simulation results are converging to Ghia's results [32]. However, as the Reynolds number increases, the resolution has to also increase to produce satisfactory results.

Fig.5.5, 5.6 and 5.7 present pressure profiles along the diagonal line, from (0.0 m, 0.0 m) to (1.0 m, 1.0 m). With a resolution of 160×160, STAR-CD can achieve almost the same accuracy for velocity as [32], and corresponding pressure results are used for validation. ISPH_DFS produces almost



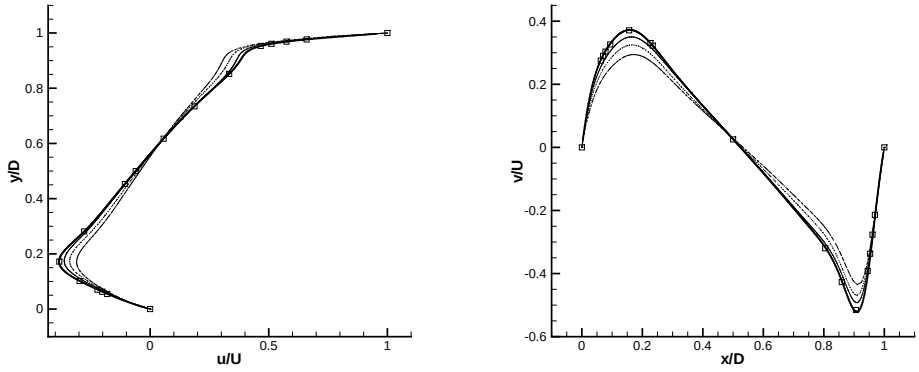
(a) (b)

Figure 5.2: Velocity profiles in lid-driven cavity, $Re = 100$. Velocity components are normalized with the lid velocity U, $U = 1$. The distances are normalized with the square length D, $D = 1$. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$. - - - = results with 41×41 resolution; $\cdots$ = results with 81×81 resolution; — = results with 161×161 resolution; Thick solid line = STAR-CD with 160×160 resolution; □ = Ghia 1982 [32].



(a) (b)

Figure 5.3: Velocity profiles in lid-driven cavity, $Re = 400$. Velocity components are normalized with the lid velocity U, $U = 1$. The distances are normalized with the square length D, $D = 1$. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$. $- - -$ = results with 81×81 resolution; $\cdots$= results with 161×161 resolution; — = results with 241×241 resolution; Thick solid line = STAR-CD with 160×160 resolution; □ = Ghia 1982 [32].



(a) (b)

Figure 5.4: Velocity profiles in lid-driven cavity, $Re = 1000$. Velocity components are normalized with the lid velocity U, $U = 1$. The distances are normalized with the square length D, $D = 1$. (a) Horizontal velocity component profile, at $x/D = 0.0m$; (b) Vertical velocity component profile, at $y/D = 0.0m$. - - - = results with 81×81 resolution; $\cdots$ = results with 161×161 resolution; — = results with 241×241 resolution; Thick solid line = STAR-CD with 160×160 resolution; □ = Ghia 1982 [32].

identical results for $Re = 100$ for all resolutions. For the higher Reynolds numbers, results can be seen to converge with increasing resolution, with a close agreement with 240×240 resolution.

With the same resolution, ISPH_DFS predicts flows less accurately than the finite volume method, which is probably caused by the first-order accuracy limitation of ISPH_DFS, while the finite volume code STAR-CD can provide second-order predictions [48, 63].

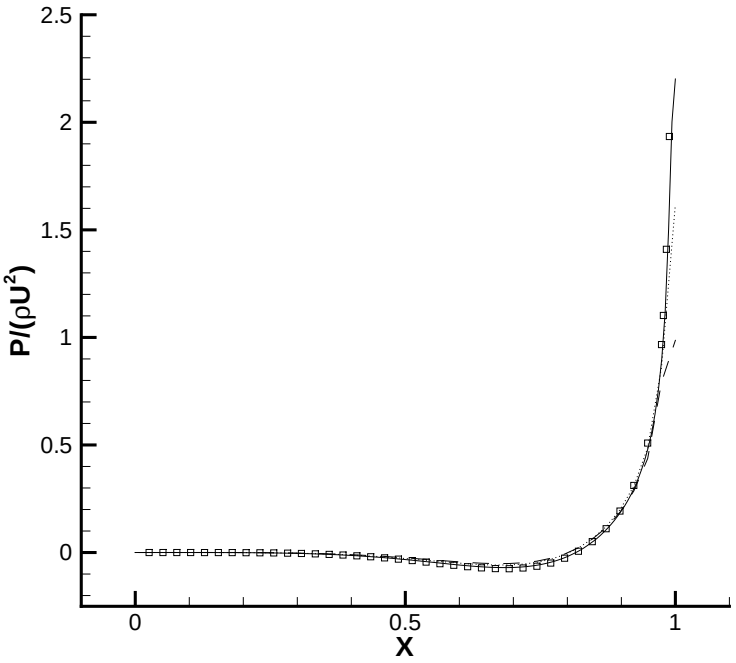


Figure 5.5: Pressure profile along the square diagonal, from (0.0, 0.0) to (1.0, 1.0), in lid-driven cavity, $Re = 100$.
□ = STAR-CD with 160×160 resolution; − − − = ISPH_DFS with 41×41 resolution; · · · = ISPH_DFS with 81×81 resolution; — = ISPH_DFS with 161×161 resolution.

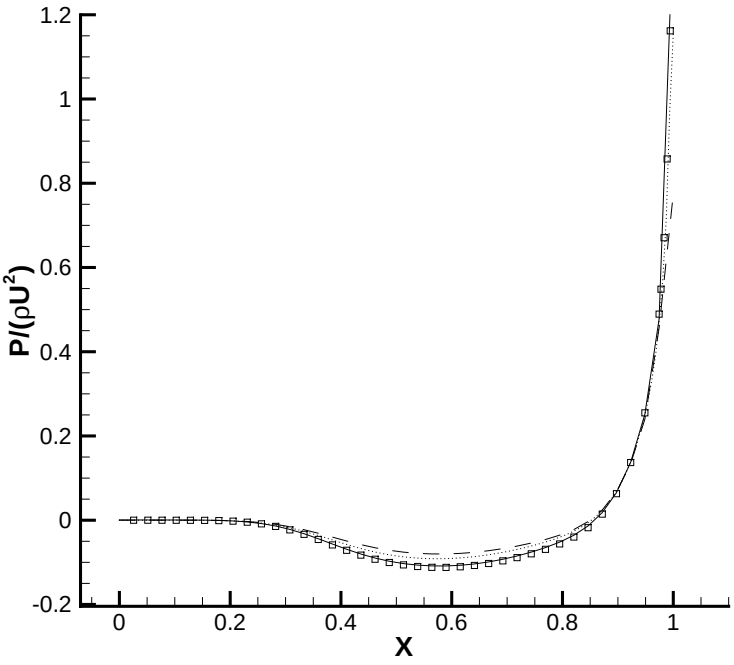


Figure 5.6: Pressure profile along the square diagonal, from (0.0, 0.0) to (1.0, 1.0), in lid-driven cavity, $Re = 400$.
□ = STAR-CD with 160×160 resolution; −−− = ISPH_DFS with 81×81 resolution; · · · = ISPH_DFS with 161×161 resolution; — = ISPH_DFS with 241×241 resolution.

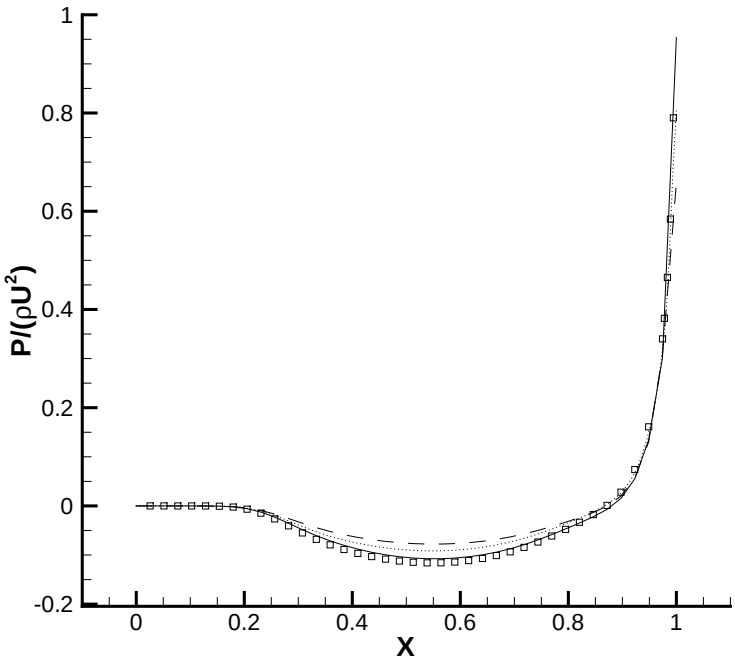


Figure 5.7: Pressure profile along the square diagonal, from (0.0, 0.0) to (1.0, 1.0), in lid-driven cavity, $Re = 1000$.
□ = STAR-CD with 160×160 resolution; −−− = ISPH_DFS with 81×81 resolution; · · · = ISPH_DFS with 161×161 resolution; — = ISPH_DFS with 241×241 resolution.

5.2 2-D laminar flow past a circular cylinder in a channel

5.2.1 Case description

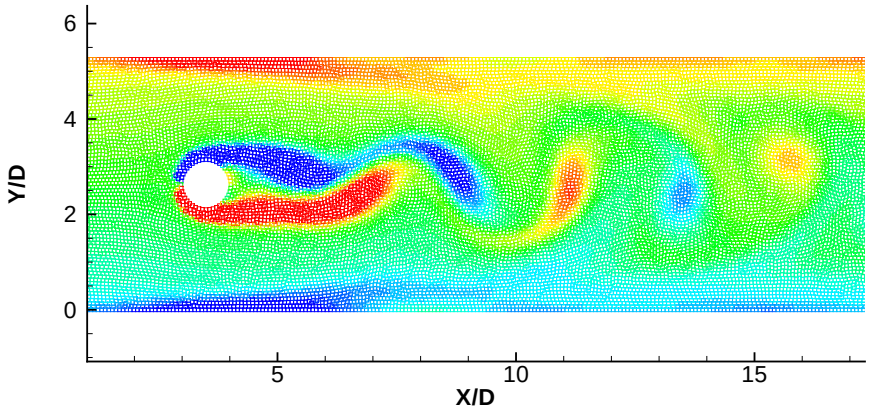


Figure 5.8: Vortex shedding around a cylinder placed in a periodic channel is simulated by incompressible SPH, $Re = 100$. The colored contour shows the vorticity.

When the fluid passes over a body, separation will happen depending on the body shape and the incident flow. Depending on the flow and fluid conditions, separation may cause vortex shedding, which occurs as vortices are created at the back of the body and become detached from each side periodically, shown in Fig. 5.8. This phenomenon happens in many engineering problems, such as buildings, bridges, offshore pipelines and even airfoils at high angles of attack etc . If the frequency of the shedding vortex matches the resonance frequency of the structure, the structure will resonate and the structure's movement can become self-sustaining, which may cause structure failure. Therefore, full understanding of this fluid behavior is very important.

δr	0.025	0.02	0.015	0.01	0.005
$Re = 20$	10466	16266	n/a	64379	256114
$Re = 100$	33426	51966	101263	n/a	n/a

Table 5.2: The particle number in each case.

The flow past a circular cylinder located in a channel is investigated to assess the ability of the code to predict drag and lift in a flow with the obstacle. Periodic boundaries are applied at the two ends of the channel. The Reynolds number is calculated as

$$Re = \frac{UD}{\nu}, \tag{5.2}$$

where U is the bulk/average velocity in the channel, D is the cylinder diameter, $D = 2R$, and ν is the kinematic viscosity, which is set as $1.0 \times 10^{-3} m^2/s$. Two situations with two different Reynolds numbers, 20 and 100, are simulated. Therefore, the bulk velocities U corresponding to two different Reynolds numbers, 20 and 100, are $0.01/R$ and $0.05/R$.

Four different initial particle sizes, δr =0.025, 0.02, 0.01 and 0.005, were used for cases with $Re = 20$, and three different initial particle sizes,δr =0.025, 0.02 and 0.015, were used for cases with $Re = 100$. Particle numbers in each cases are listed in Table 5.2.

The geometry of the case is shown in Fig. 5.9. To allow the flow to become fully developed, the channel is much longer for the case with $Re = 100$. The geometry parameters in different Reynolds situations are listed in Table 5.3.

Reynolds Number Re	L/R	W_{ch}/R	$R\,(m)$	l/R
20	64	10	0.1	7
100	204	10	0.1	7

Table 5.3: Geometry parameters in bluff body cases.

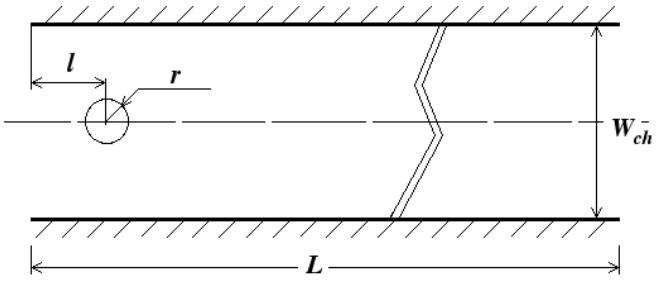


Figure 5.9: Geometry of the bluff body test case.

5.2.2 Code Saturne

The prediction of the velocity and pressure field from ISPH method are compared with code Saturne, which is a finite volume code developed in R & D in Electricite de France (EDF) [1]. The same geometry size and boundary conditions are used in F.V. simulations. Three different meshes with different resolutions, 16858 cells, 67353 cells and 255711 cells, are used for the case with $Re = 20$, while another three different meshes with different resolutions, 15914 cells, 62094 cells and 251293 cells, are used for the other case with $Re = 100$. One of the meshes with a resolution of 67353 in the case with $Re = 20$ is shown in Fig. 5.10. In simulations by code Saturne, the SIMPLE

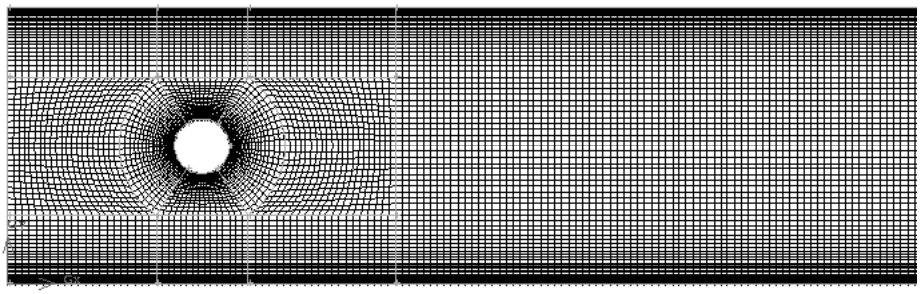


Figure 5.10: Meshes used in the FV simulations.

algorithm is used for the pressure-velocity coupling. Second-order central differencing scheme is used for the convection term. Convergence criterion is set to 10^{-6} for linear solvers in the internal iterations.

5.2.3 Simulation results and discussion

Re=20

Fig. 5.11 shows the velocity and pressure contour in the case with $Re = 20$ at the time when the flow is fully accelerated. The new ISPH approach provides very smooth prediction for both velocity and pressure field. Fig. 5.12 and 5.13 present the velocity and pressure profiles, at $Re = 20$, along the cross sections, $x/D = 2.5$, 5.0, 7.5, 10.0, 12.5 and 30.0. Both the results from ISPH with an initial particle distance $\delta r = 0.01$ and code Saturne with a grid number of 255711 are plotted. The vertical coordinates are normalised by

the cylinder diameter D, the velocity by the bulk/average velocity U and the pressure by $\frac{1}{2}\rho U^2$. To better illustrate the profiles change along the channel, excluding the profile at $x/D = 2.5$, normalised variable values are shifted with the cross-section coordinates x, for example, the normalised horizontal velocity u/U is changed by $u/U + x$, for $x = 1.0$. It can be observed in Fig. 5.12 (a) that due to the blockage effect of the cylinder, the horizontal velocity is greatly reduced in the middle of the channel, and at the upstream of the cylinder the vertical-velocity profiles show both positive and negative peaks at upper and bottom halves of the channel. The positions of the positive and negative peaks will be swapped downstream of the cylinder due to the flow recirculation. The blockage effect is reduced as the fluid moves along the channel. Both code Saturne and ISPH give almost identical predictions for the velocity and pressure field.

Other noteworthy variables to predict in this bluff body case are the lift and drag forces. They are calculated as

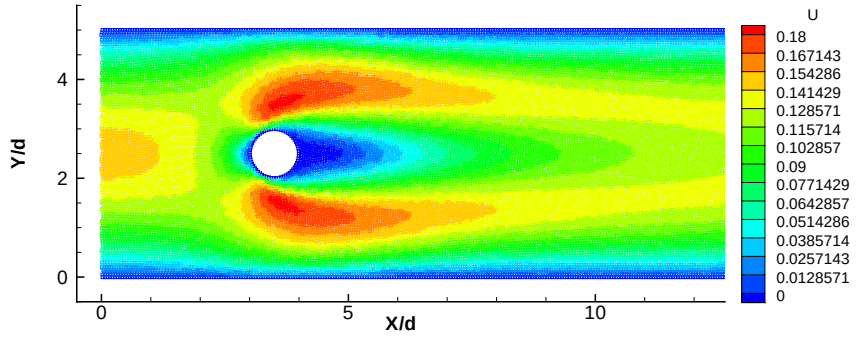
$$F_d = \int_\Gamma p n_x - \int_\Gamma (\tau_{xx} \cdot n_x + \tau_{xy} \cdot n_y) \tag{5.3}$$

$$F_l = \int_\Gamma p n_y - \int_\Gamma (\tau_{yx} \cdot n_x + \tau_{yy} \cdot n_y) \tag{5.4}$$

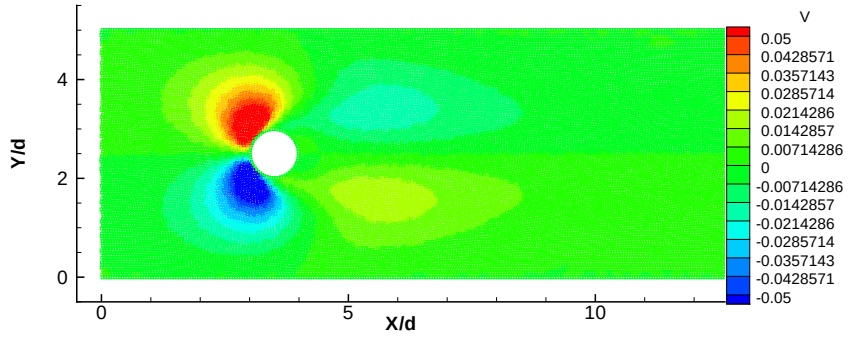
where n_x and n_y are two components of the normal direction at the cylinder surface; τ_{xy} is the viscous stress, $\tau_{xy} = \mu \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right)$. The forces are usually normalised by the bulk/average velocity U as

$$C_d = \frac{F_d}{0.5\rho U^2 D} \tag{5.5}$$

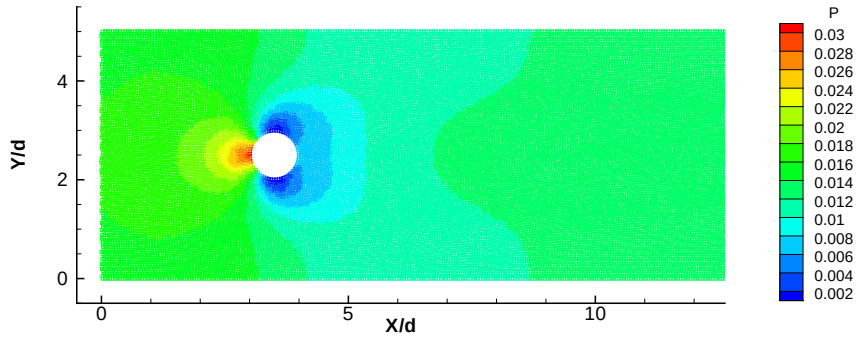
$$C_l = \frac{F_l}{0.5\rho U^2 D} \tag{5.6}$$



(a)

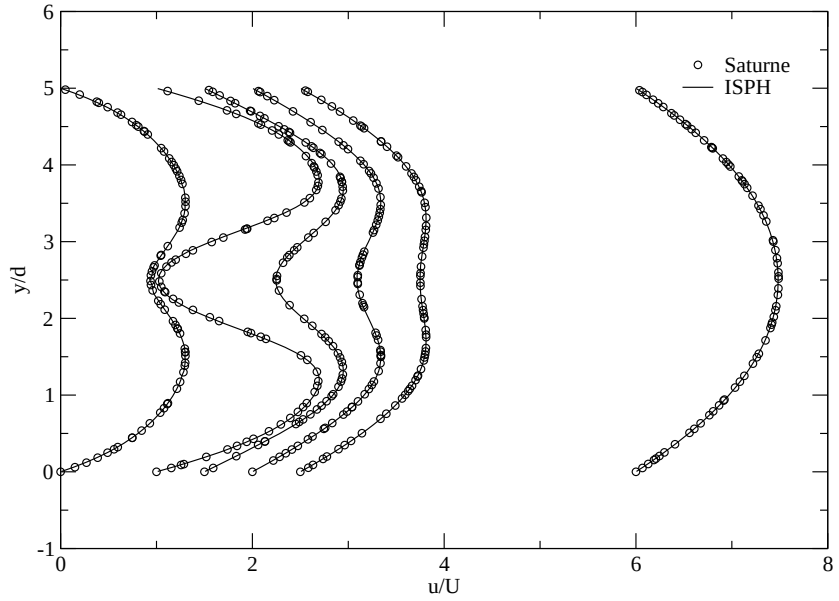


(b)

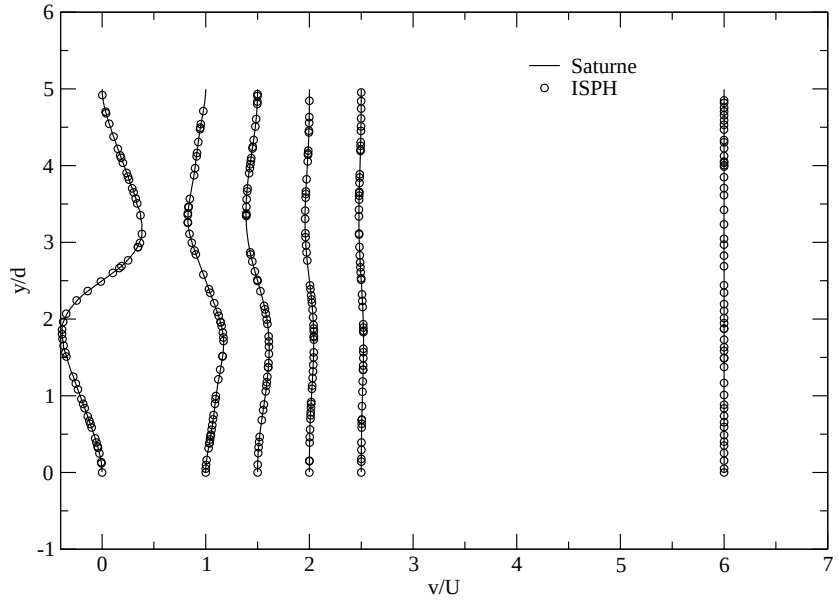


(c)

Figure 5.11: Velocity and pressure field in the bluff body case, $Re = 20$, initial particle size $\delta r = 0.01$, at the time when the fluid fully developed. (a) Contour of the horizontal velocity U; (b) Contour of the vertical velocity V; (c) Contour of the Pressure P.



(a)



(b)

Figure 5.12: The velocity profiles along different cross sections, $x/D = 2.5$, 5.0, 7.5, 10.0, 12.5 and 30.0, in the channel, $Re = 20$. The vertical coordinates y and velocity components u and v are normalised by the cylinder diameter D and the bulk velocity U. To better present the profiles along the channel, excluding the profile at $x/D = 2.5$, normalised velocity values u/U and v/U are shifted with the cross-section coordinates x.
(a) Profile of the horizontal velocity u; (b) Profile of the vertical velocity v.

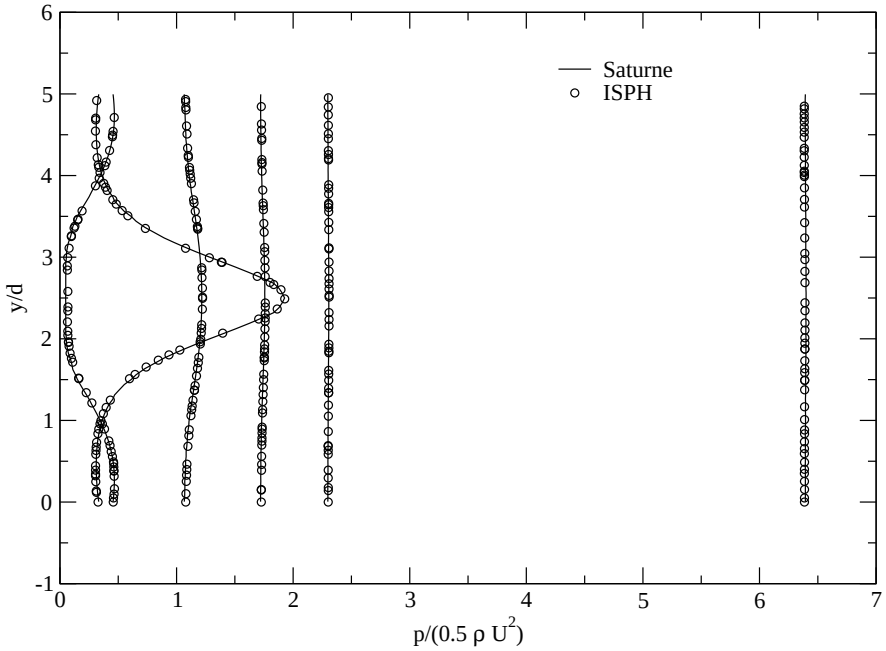


Figure 5.13: The pressure profiles along different cross sections, $x/D = 2.5$, 5.0, 7.5, 10.0, 12.5 and 30.0, in the channel, $Re = 20$. The vertical coordinates y and pressure P are normalised by the cylinder diameter D and $\frac{1}{2}\rho U^2$ respectively. To better present the profiles along the channel, excluding the profile at $x/D = 2.5$, normalised pressure values $p/\frac{1}{2}\rho U^2$ is shifted with the cross-section coordinates x.

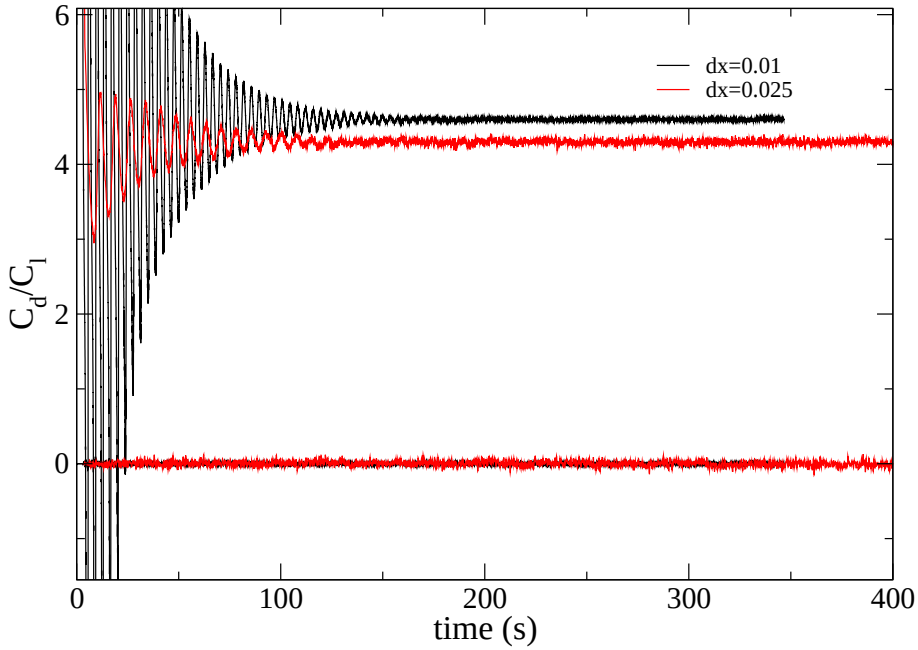
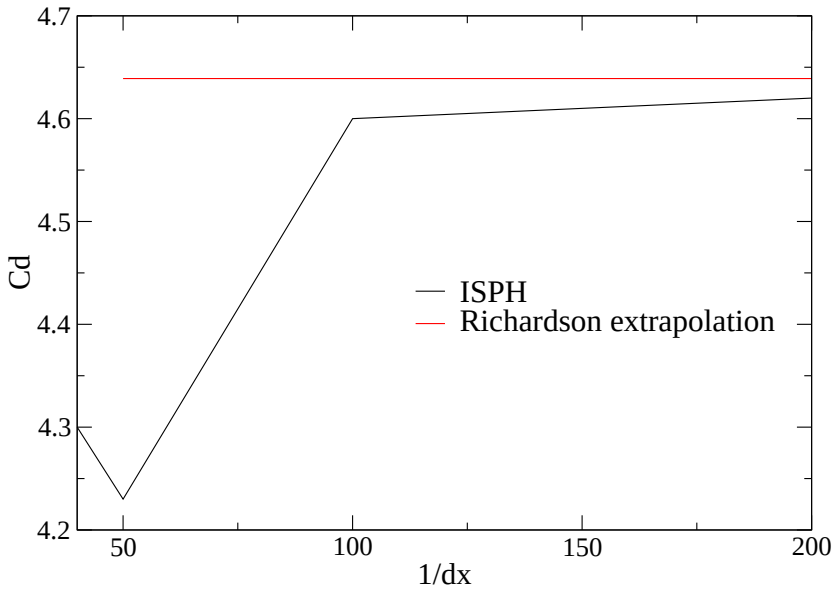


Figure 5.14: Drag and lift coefficients results in ISPH simulations.

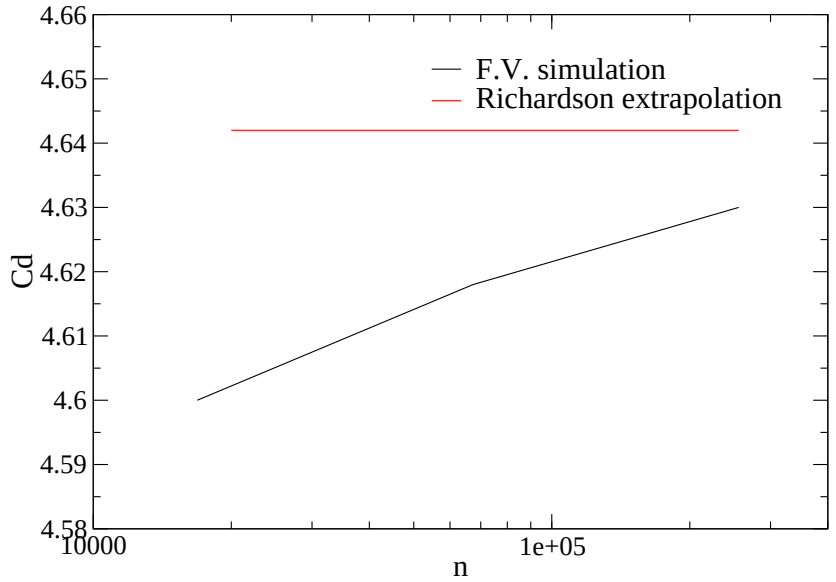
where D is the cylinder diameter. In ISPH simulations, different initial particle sizes, $\delta r = 0.025$, 0.02, 0.01 and 0.005, are used for $Re = 20$. It should be pointed out that due to the movement of particles, which introduces small variance in the interpolation, the results of drag and lift force are not as smooth as counterparts in the finite-volume simulations, shown in Fig. 5.14. After the fluid is fully accelerated, the time-averaged values are calculated for the coefficients. In the case with $Re = 20$, the values for C_d with different resolutions and extrapolated values, based on Richardson extrapolation, are plotted in Fig. 5.15. The extrapolated values, 4.640 for ISPH and 4.642 for Saturne, are calculated.

Re=100

In the case with $Re = 100$, three different initial particle spacings, $\delta r = 0.025$, 0.02 and 0.015, are tested here. As different shedding frequencies predicted



(a) Drag coefficients change against increasing resolution 1/dx



(b) Drag coefficients change against increasing mesh number n

Figure 5.15: Drag coefficients C_d from both ISPH and code Saturne as a function of resolution, $Re = 20$. dx in (a) is the initial particle size. An extrapolated value has been calculated.

and different time steps used in both ISPH and FVM, it is difficult to compare contour graphs at a certain time. In Fig. 5.16, the contours of velocity and pressure fields from ISPH are plotted, and it is shown in the graphs that the new ISPH approach provides smooth predictions for the velocity and pressure fields. Fig. 5.17 and 5.18 show velocity and pressure profile comparisons between FVM and ISPH. Due to the anisotropic and changing particle distribution in ISPH, calculating the time-averaged value is not achieved here. Therefore, all the velocity and pressure values within several shedding periods along the cross sections are plotted in the same graph as time-averaged velocity profiles from code Saturne. It can be seen from Fig. 5.17 and 5.18 that ISPH method gives similar velocity and pressure predictions to code Saturne. The maximum and minimum values for C_d and C_l from code Saturne are also presented in Fig. 5.19. The extrapolated values for maximum and minimum drag coefficients are 2.829 and 2.673, while the lift coefficients are ± 1.215. Although the averaged values are quite close to the predictions by code Saturne it is difficult for ISPH to provide smooth profiles for the drag and lift coefficients as particles moving around the cylinder causes inaccuracy and noise in the force calculations, as shown in Fig. 5.20. A clear vortex shedding period, $T \simeq 1.54s$, is obtained from the finest resolution, as presented in the zoomed part in Fig. 5.20. Comparing with the counterpart from code Saturne, $T = 1.68s$, there is 8% relative difference. The Strouhal numbers, $St = \frac{fD}{U}$, are calculated for both methods based on the bulk velocity, U, and cylinder diameter, D. They are listed in Tab. 5.4. Because the maximum and minimum drag coefficients are difficult to obtain from ISPH simulations, and it is difficult to run another simulation with a higher resolution on a single CPU, the averaged drag coefficient, $C_d = 2.34$, is

	Strouhal No.	Averaged Drag Coe. (C_d)	Lift Coe. (C_l)
ISPH	0.2597	2.34	± 0.87
Code Saturne	0.2381	2.751	± 1.215
Relative Diff.	9%	15%	28.4%

Table 5.4: Strouhal numbers, drag and lift coefficients from ISPH and FVM, $Re = 100$. FVM simulations are conducted with open source code, Code_Saturne [1].

calculated for ISPH simulation; the maximum and minimum lift coefficients from the ISPH simulation with $\delta r = 0.015$ are approximately ± 0.87. The relative differences between results from ISPH and code Saturne are 15% for the drag coefficient and 28.4% for lift coefficient.

It should be noted here that accurate predictions are obtained in [81] by using local refinement around the bluff body and fixing all particles or particles around the bluff body with the help of an Arbitrary Lagrangian-Eulerian (ALE) [41] type algorithm. The numerical error introduced by the irregular particle distributions, which appear in SPH simulations with moving particles, is avoided in [81]. Also the local refinement around the cylinder improves the accuracy of force predictions. However, without an ALE-type ISPH algorithm, if particle positions are fixed and resolutions are refined around obstacles, we need remeshing and conservative interpolations with various support sizes which are still an unconcluded research topic now [9], and also the meshless advantage is lost with all particle fixed. Therefore, we did not extend the work to regular and fixed particle distribution. However, wall boundary methods with smaller numerical errors are needed in the future.

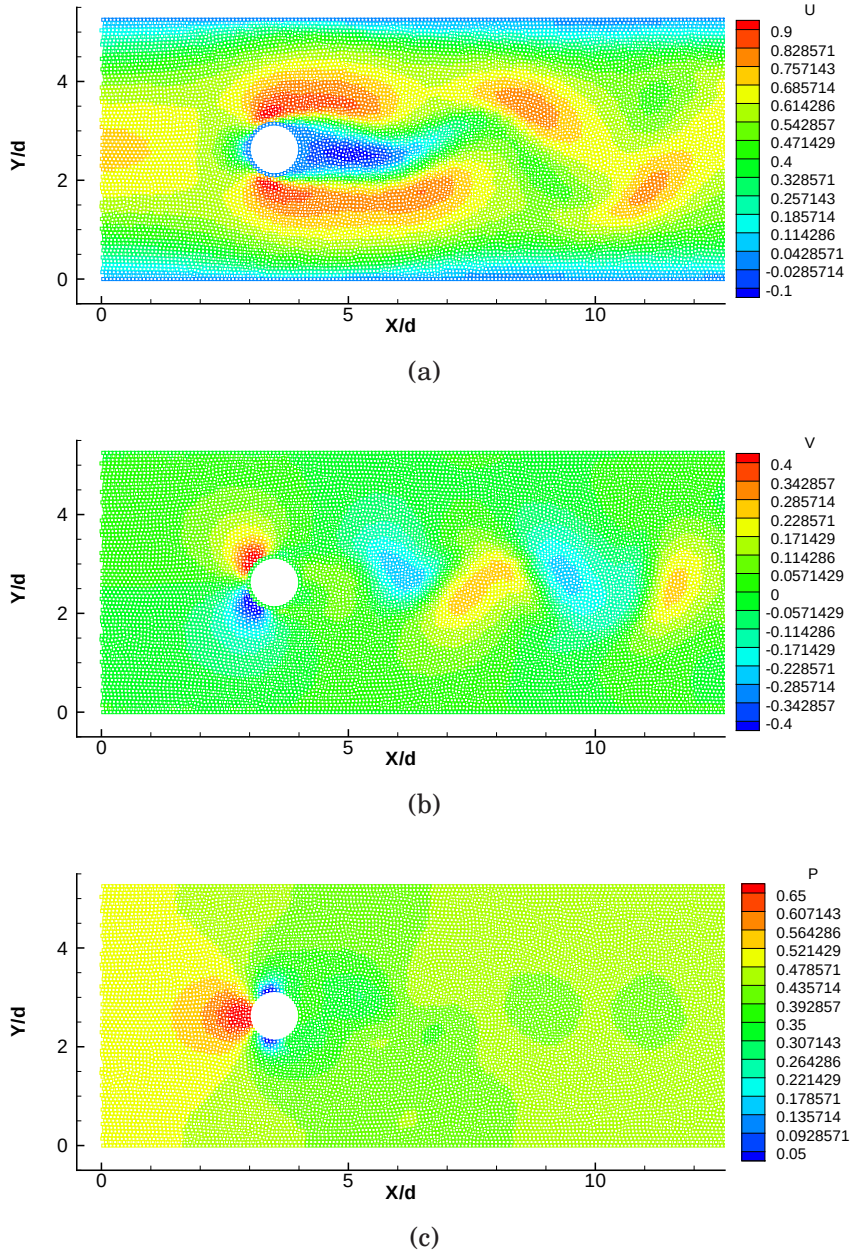
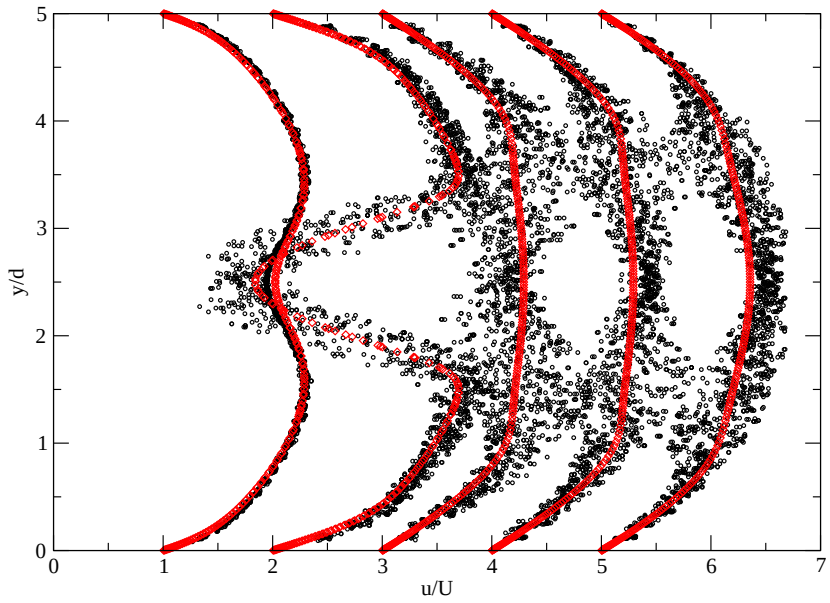
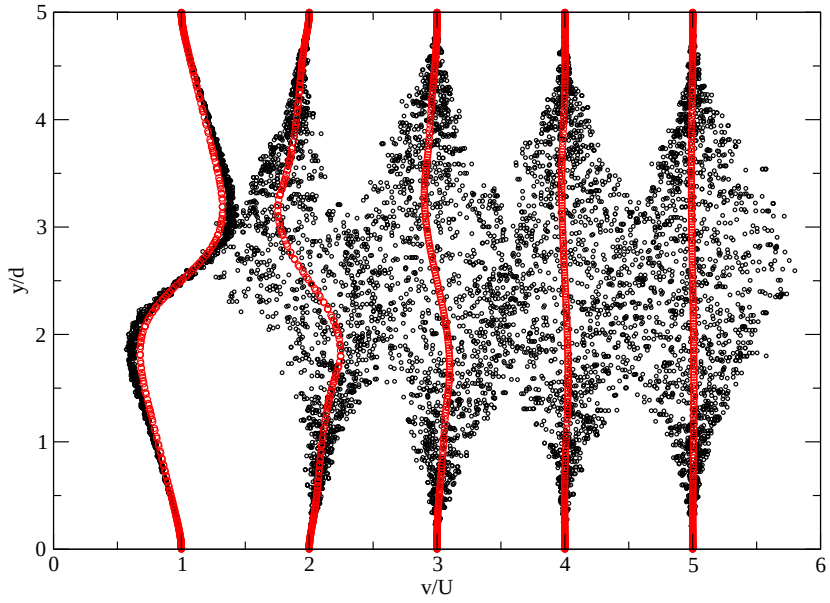


Figure 5.16: Velocity and pressure field in the bluff body case, $Re = 100$, initial particle size $\delta r = 0.015$, at $time = 196.65\,s$. (a) Contour of the horizontal velocity U; (b) Contour of the vertical velocity V; (c) Contour of the pressure P.



(a)



(b)

Figure 5.17: The time-averaged velocity profiles along different cross sections, $x/D = 2.5$, 5.0, 7.5, 10.0 and 12.5, in the channel, $Re = 100$. The vertical coordinates y and velocity components u and v are normalised by the cylinder diameter D and the bulk velocity U. To better present the profiles along the channel, normalised velocity values u/U and v/U are shifted with the cross-section coordinates $x/0.5$. Red dots: F.V. method; black dots: ISPH method. (a) Profiles of the horizontal velocity u; (b) Profiles of the vertical velocity v.

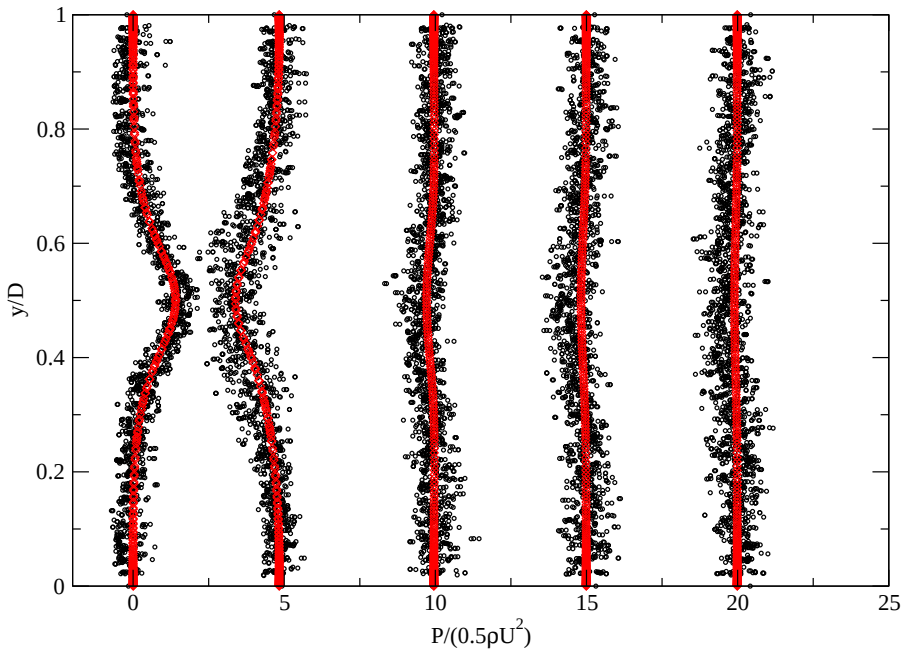
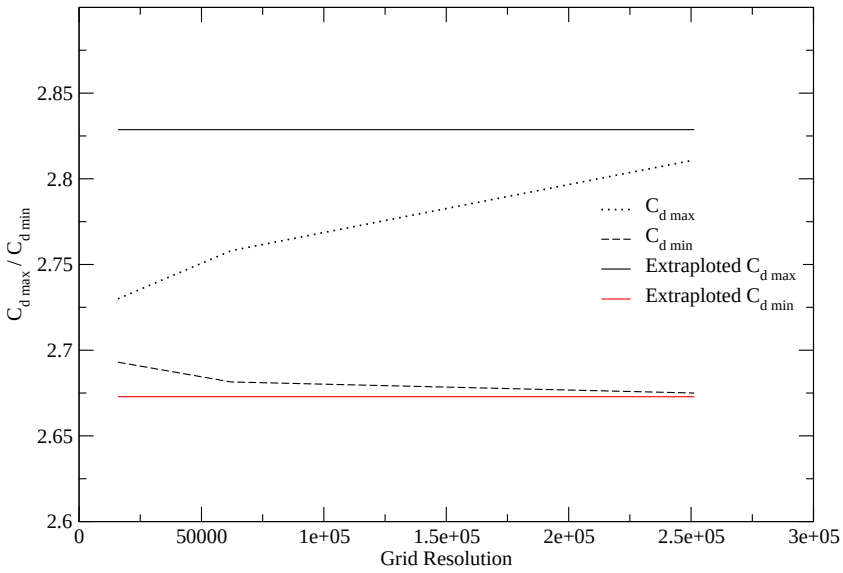
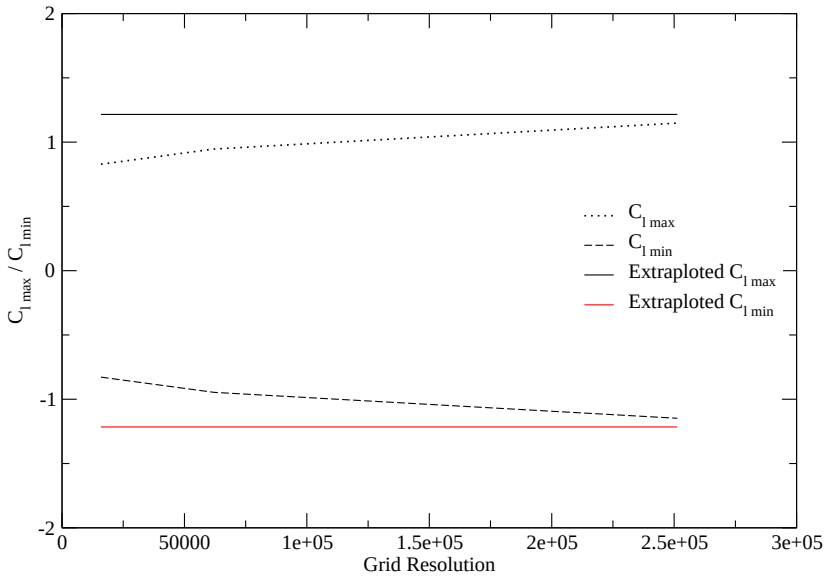


Figure 5.18: The pressure profiles, from left to right, along different cross sections, $x/D = 2.5$, 5.0, 7.5, 10.0 and 12.5, in the channel, $Re = 100$. The vertical coordinates y and pressure P are normalised by the cylinder diameter D and $\frac{1}{2}\rho U^2$ respectively. To better present the profiles along the channel profiles are shifted. Red dots: F.V. method; black dots: ISPH method.

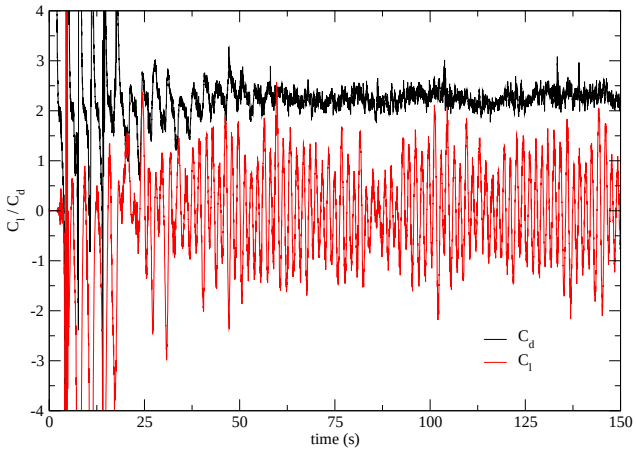


(a) Maximum and minimum drag coefficients change against increasing grid number

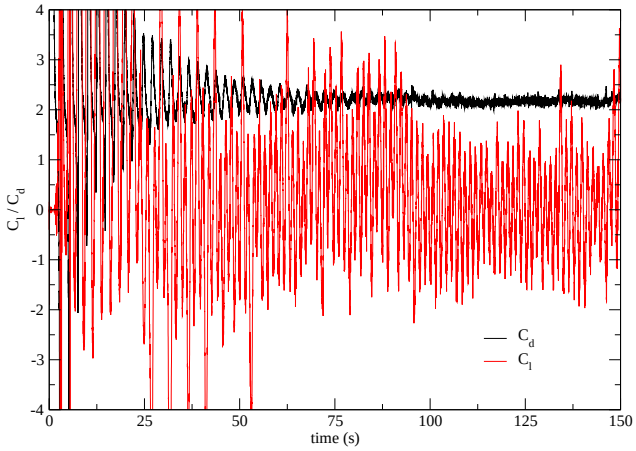


(b) Maximum and minimum lift coefficients change against increasing mesh number n

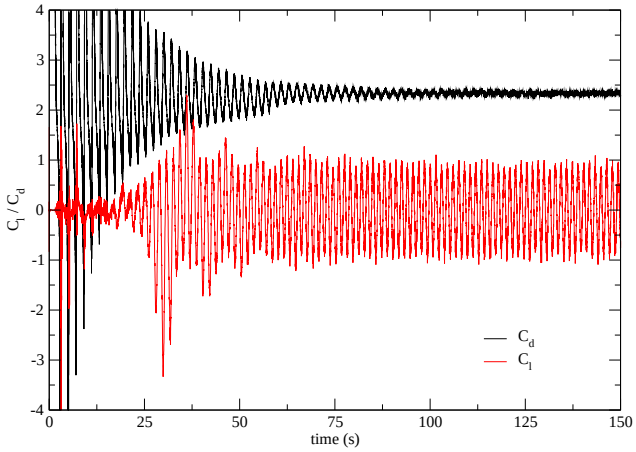
Figure 5.19: Drag C_d and lift C_l coefficients from code Saturne with different resolutions, $Re = 100$. An extrapolated value has been calculated.



(a)



(b)



(c)

Figure 5.20: Drag C_d and lift C_l coefficients from ISPH with different resolutions, $Re = 100$. **(a)** $dx = 0.025$; **(b)** $dx = 0.02$; **(c)** $dx = 0.015$.

5.3 Partial conclusion

Through the benchmark test cases, lid-driven cavity and bluff body, the algorithm and the code have been validated. Both cases are simulated by both ISPH and FVM, with the commercial software STAR-CD or code Saturne [1].

Three different Re situations, $Re = 100$, 400 and 1000, have been simulated for the lid driven cavity. However, it has been shown that the ISPH method provides a slower convergence rate, compared to second-order of FVM, as shown by the velocity profile comparison in Fig. 5.4. The pressure predictions from ISPH match the converged finite-volume results very well.

For the bluff body, cases with two different Reynolds numbers are simulated here. The vortex shedding happened at $Re = 100$. The lift and drag coefficients are calculated on both ISPH and FVM with code Saturne [1]. At $Re = 20$, both methods give almost identical predictions. However, differences appear at higher Reynolds number, $Re = 100$. The ISPH prediction of force coefficients is quite noisy with low resolutions. The vortex-shedding period is only obtained from the highest resolution in ISPH. The time-averaged drag coefficient is calculated from ISPH simulation with the highest resolution. Relative differences between ISPH and code Saturne, 8%, 15% and 28.4%, are obtained for vortex-shedding period, drag and lift coefficients respectively. Wall boundary methods with less numerical errors are needed in the future.

Chapter 6

ISPH_DFS Method in Free-Surface Flow Simulations

TRaditional Eulerian methods have some difficulties in describing the free surface, while Lagrangian methods possess natural ability to describe the details of complicated phenomena. It is much easier for Lagrangian method to include complicated integration. The traditional WCSPH method has predicted some highly transient flows quite well [96, 18, 115, 116, 28], pressures were noisy and the method highly dissipative [63]. Therefore, the application of ISPH method in free-surface prediction is desirable.

A highly accurate, incompressible, mesh-free, noise-free method for arbitrary free-surface flows is attractive for many problems in engineering which involve both fluid-structure interaction and multi-phase fluid simulations. Rigorous validation is desirable against analytical or highly accurate solutions as undertaken for internal flows. The following phenomena can occur in one simulation or in different simulations separately:

1. impulsive fluid-structure interaction,

2. free-surface evolution with very high curvature,

3. wave propagation.

We use the analytical solution for impulsive plate motion with zero gravity [86] which shows asymptotic-like behavior at the fluid-solid intersection. The dam-break flow has received considerable attention but has not been fully exploited. For the wet-bed case we use the highly accurate solution for flow acceleration at zero time, with a singularity in the free surface at the lower gradient discontinuity [102]. With an analytic initial condition a highly accurate high-order boundary integral method for potential flow [24] is available for small times showing jet-like mushroom behavior [102]. This is preferable to direct experimental comparison, which shows similar phenomena, since the plate removal time is of the same order as the time scale for the formation of the initial flow structures. Finally non-dissipative wave propagation should be demonstrated. Regular waves are generated by a piston-type paddle for several periods and surface profiles, velocities and pressures are compared against accurate stream-function theory [94]. This is undertaken for small and moderate waves for which linear wavemaker theory is realistic. To the authors' knowledge, this is the first time ISPH simulations of free-surface flows have been validated for both pressure and velocity fields.

In this Chapter, without special declaration, the ISPH method used is ISPH_DFS method introduced in Chapter 2.

6.1 Identification of free-surface particles

In free-surface simulations, the free-surface boundary is located simply through the value of $\nabla \cdot \mathrm{r}$ without the kernel correction. If the particle has a full kernel support, the value of $\nabla \cdot \mathrm{r}$ from SPH interpolation should be around 2.0, without the interpolation error. For particles close to the free surface, this value should be smaller than 2.0. In the free-surface simulations, the criterion to locate free-surface particles is set as 1.5. More details can be obtained in [63]. However, it is found in the simulations that this method cannot locate the free surface with absolute precision. Some internal fluid particles may have values of $\nabla \cdot \mathrm{r}$ smaller than the criterion, while some free-surface particles may have larger values. This misjudgment for the free-surface boundary will introduce certain error in the results, although this appears to be insignificant in this work. To keep the free-surface prediction uncontaminated by the artificial movement of particles, the shifting in §2.2.4 is not applied for the free-surface particles.

6.2 Truncated-kernel error on the free surface

As the kernel function is an approximation of the Dirac δ function, full kernel support is needed to give accurate kernel interpolation. However, particles at and adjacent to the free surface cannot obtain the full kernel support, which introduces error in the free-surface prediction, potentially causing non-physical instability. The truncated-kernel error in SPH interpolation is also reported in [26]. Even with the improved Laplacian operator, Eq. 2.21, Schwaiger has shown a large relative error to the analytical value at the free

surface, almost 70% [99]. This error strongly distorts the free surface, and sometimes hinder the convergence of the prediction.

The same numerical test as that in [99] has been repeated here, where the values for $\nabla \cdot \nabla\varphi$ with the test function $\varphi = x^m + y^m$, m =2, 3 and 4, are calculated respectively with the operator in [77], referred to as Laplacian operator proposed by Morris et al. (LM) here, and the counterpart in [99], referred to as Laplacian operator proposed by Schwaiger (LS) here. To avoid singularity in the denominator of the relative error and to keep the same geometry as in [99], a uniform 50×50 particle distribution is used within a unit square with a dimension $2.0 \leq x/D \leq 3.0$ and $2.0 \leq y/D \leq 3.0$. The relative error, defined as

$$\varepsilon = \frac{\mid \phi_{SPH} - \phi_{exact} \mid}{\phi_{exact}}, \tag{6.1}$$

where ϕ_{SPH} is the value obtained from SPH calculation, and ϕ_{exact} is the analytical value, has been plotted in Fig. 6.1. Fig. 6.1 shows the same results as in [99]: the relative errors for the LS operator on the free surface are around 70%, while the relative errors for the LM operator on the free surface reaches 4000%; at the row just below the free surface, the relative error for LS operator rapidly reduces to around 4%, while LM operator still gives a high relative error value, around 500%. In this work, the LS operator is used for viscous and Laplacian terms. A similar test is also conducted here for gradient operator introduced in §2.1.2 on the same particle distribution. The test function is $\varphi = y^m$, with m =1, 2 and 3. The relative error for $\frac{\partial \varphi}{\partial y}$ along the line $x/D = 2.5$ is plotted in Fig. 6.2. The maximum relative error, located at the free surface, is around 1% with the gradient operator for

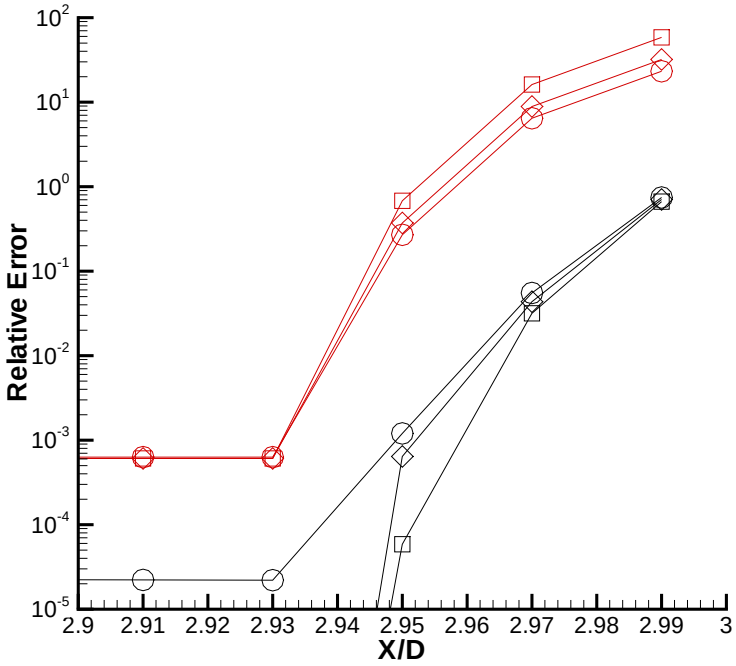


Figure 6.1: Relative error profiles at the right end along $y/D = 2.5$ for test function $\varphi = x^m + y^m$ with the Laplacian operator in [99] and that in [77], where φ is the general function; m are equal to 2, 3 and 4; D is the domain length, equal to 1. Profiles for the Laplacian operator in [99] are black, while profiles for the Laplacian operator in [77] are red. The same symbols are used for the same test functions φ. Circles : test function with $m = 2$; diamonds : test function with $m = 3$; squares : test function with $m = 4$.

the test function, and for function $\varphi = y$, the accuracy has reduced to the machine precision.

Because the pressure value on the free surface is constant, conveniently zero, the Poisson equation is not solved for the free-surface particles, and the error introduced by the Laplacian operator on the free surface thus thus does not strongly contaminate predictions. However, under different particle distribution situations, the error may change. To investigate the influence of

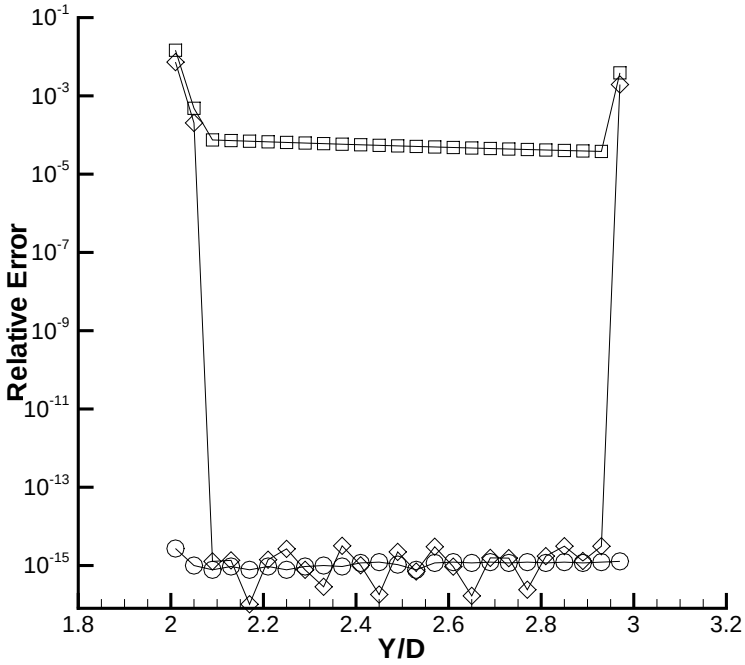


Figure 6.2: Relative error profile for $\dfrac{\partial \varphi}{\partial y}$ along $x/D = 2.5$ for test function $\varphi = y^m$ with the gradient operator introduced in §2.1.2, where φ is the general function; m are equal to 1, 2 and 3; D is the domain length, equal to 1. Circles : test function with $m = 1$; diamonds : test function with $m = 2$; squares : test function with $m = 3$.

the truncated-kernel error introduced by Laplacian operator in [99], the solutions for a set of Poisson equations, $\triangle\varphi = y^m$ with m =0, 1, 2 and 3 respectively, in a 1-D periodic open channel with a horizontal bed at $y = 0.0$ and a stationary free surface at $y = 1.0$ are calculated for a uniform particle distribution. This open channel geometry is set up by several columns of particles with periodic boundaries at two ends. $\triangle\varphi = 1$ with the same boundaries is also solved for a non-uniform particle distribution. In the open channel, the Dirichlet boundary condition $\varphi = 1.0$ is applied at the free surface $y = 1.0$, and the Neumann boundary $\frac{d\varphi}{dy} = 10.0$ at $y = 0.0$. With a uniform particle distribution, the relative errors are plotted in Fig. 6.3. For all the exponent numbers considered, the relative errors are no larger than 7×10^{-5}, and exactly 0 on the free surface. However, the relative errors could be greater with the non-uniform particle distribution. In further tests, particles above 0.8 are shifted in both x and y coordinates with small random numbers between $\pm 0.5dx$ and between $\pm 0.1dx$, where dx is the mean particle spacing. Fig. 6.4 (a) and 6.4 (b) show the relative error values along the channel for the solution of the Poisson equation $\triangle\varphi = 1.0$. It can be observed that the maximum error around the free surface reaches almost 17% in Fig. 6.4 (a) and 1.5% in Fig. 6.4 (b). It is also reported in [2], [90] and [99] that the disordered particle distribution will complicate the convergence behavior of operators. This lack of convergence caused by truncated-kernel errors on the free surface will result in noisy free-surface evolution which will be demonstrated later. Therefore, suppressing this free-surface instability is desirable but of course this may sacrifice accuracy and this needs to be investigated.

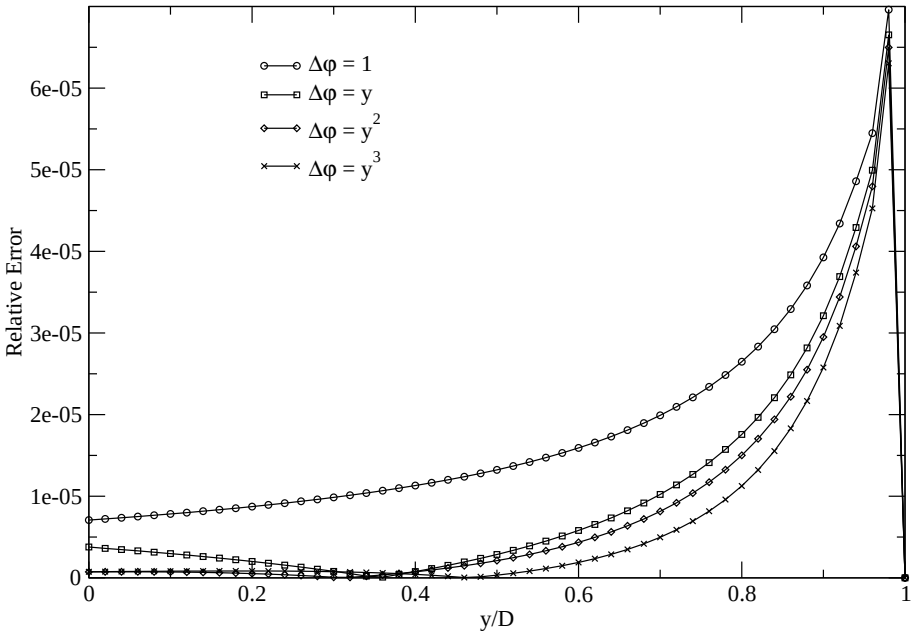
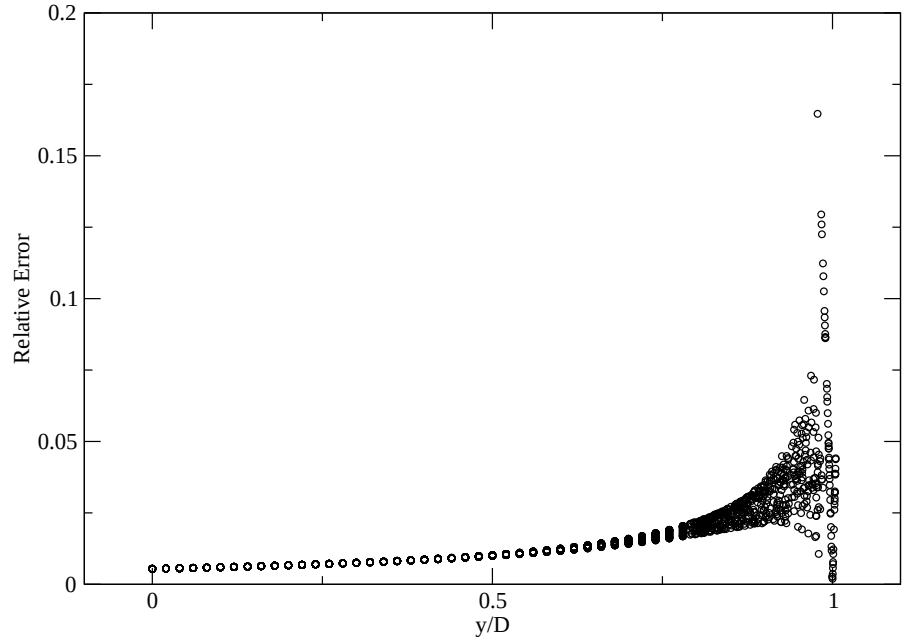


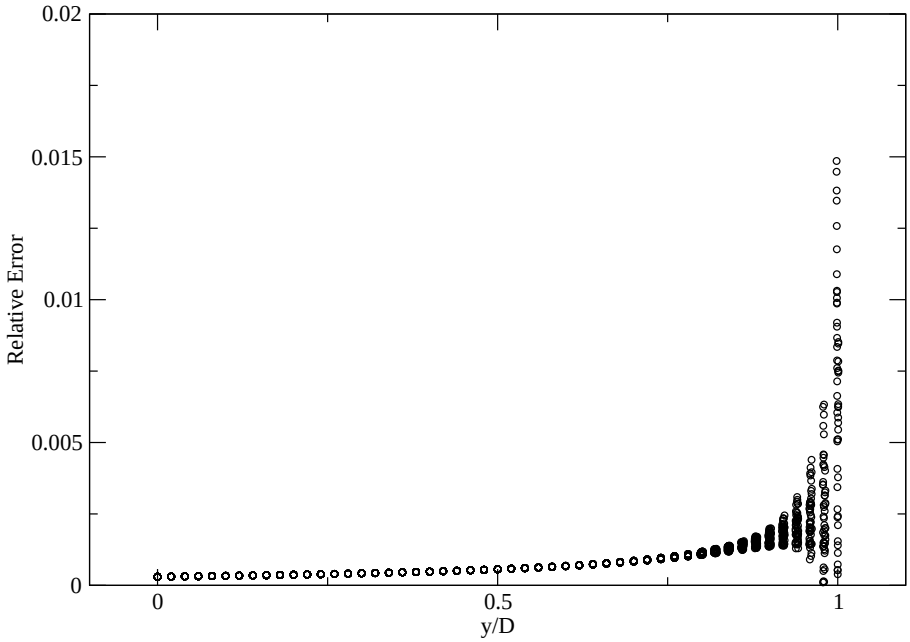
Figure 6.3: Relative errors in the solution for $\triangle\varphi = y^m$ in a 1-D periodic open channel with boundary conditions $\varphi = 1.0$ at $y = 1.0$ and $\dfrac{d\varphi}{dy} = 10.0$ at $y = 0.0$ with a uniform particle distribution, where $m = 0,\ 1,\ 2,\ 3$ in the tests.

6.3 Free-surface instability damping

It has been observed in simulations that the noisy free surface can be smoothed by artificially increasing viscosity, which is shown in Fig.6.5 as below, where the smooth free surface is predicted in the mud-flow simulation with higher viscosity, $10^{-3}m^2/s$. Here we are trying to increase the viscosity of the free-surface particles and particles adjacent to the free surface, within a distance of twice of the initial particle spacing, in order to obtain the same effect as the high viscosity used in mud-flow simulations without strongly influencing the free-surface predictions. The global maximum Peclet number $Pe_{max} = \dfrac{u_{max}dx}{\nu_d}$, with u_{max} as the maximum global velocity and dx as the initial particle spacing, is introduced here to calculated the higher viscosity ν_d around the free surface; that is, $\nu_d = \dfrac{u_{max}dx}{Pe_{max}}$. the The effect on the



(a)



(b)

Figure 6.4: Relative errors in the solution for $\triangle\varphi = 1$ in a 1-D periodic open channel with boundary conditions $\varphi = 1.0$ at $y = 1.0$ and $\dfrac{d\varphi}{dy} = 10.0$ at $y = 0.0$ with non-uniform particle distributions. (a) small random particle displacement between $\pm 0.5dx$; (b) small random particle displacement between $\pm 0.1dx$.

accuracy of free-surface predictions needs to be explored. This numerical treatment is called free-surface damping here.

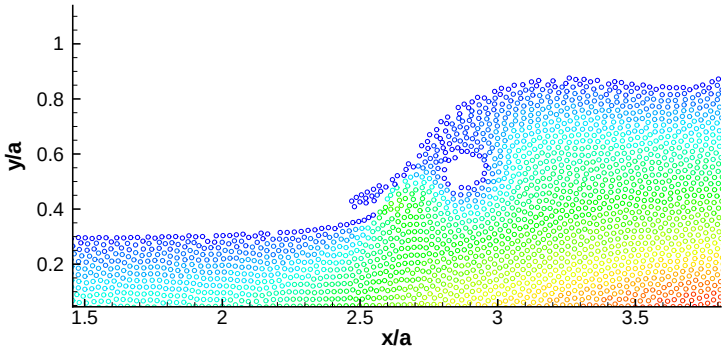


Figure 6.5: High viscosity reduces noise in the collapse of a mud column with a dimension of $a \times 2a$, where a is the dimension unit.

To compare the effect of higher viscosity on free-surface profiles and hydrodynamic variables, the collapse of water column with a dimension $a \times 2a$, and $a = 0.1m$, has been simulated. The initial water column is placed within a tank with a dimension of $4a \times 4a$. The pressure field and free surface are compared for cases with and without free-surface damping. The geometry of the case is shown in Fig. 6.6. Fig. 6.7 shows the free surfaces at the same time with and without free-surface damping. It can be observed that with damping the predicted free surface is smooth compared with the simulation without viscosity damping. It should be recognized that higher viscosity not only limits the unbounded numerical error to a small value, but also damps small-scale physical motion of the fluid. However, by choosing the damping viscosity appropriately as discussed below, this numerical tool will be shown not to reduce accuracy in the test cases considered.

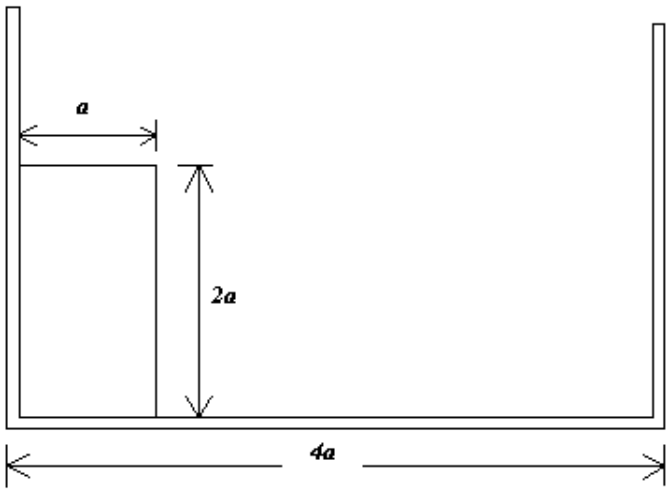
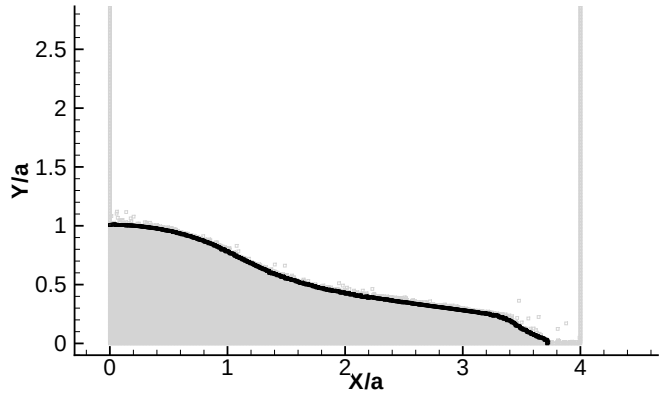
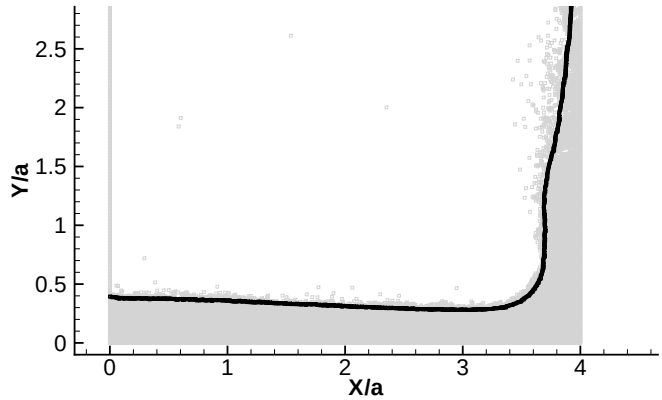


Figure 6.6: Geometry of dam break, with $a = 0.1m$.

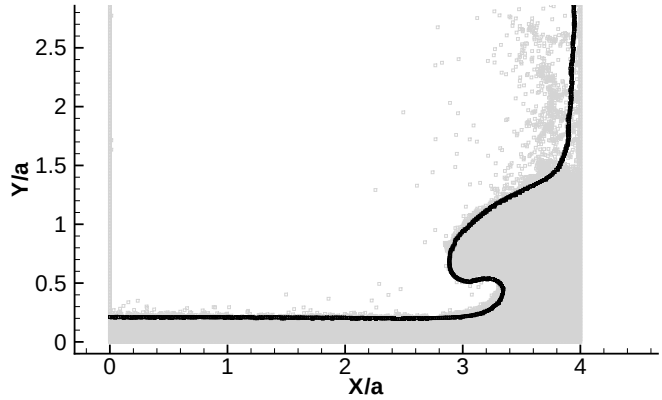
A numerical experiment has thus been conducted to investigate the relationship between the particle size, velocity and the damping viscosity. Different resolutions have been used with different viscosity values for the collapse of water column shown above. The maximum fluid velocity at the free surface, u_{max}, is estimated as $\sqrt{4ga}$ [72, 63]. The maximum Peclet number, $Pe_{max} = \dfrac{u_{max}dx}{\nu}$, is calculated through the free surface evolution for each case. If the surface viscosity is too high, the artificial diffusion will influence the free-surface prediction, while values which are too low cannot provide enough artificial damping for the truncated-kernel error. It has been shown through numerical experiments that the viscosity value should be limited to satisfy the maximum local Peclet number approximately in the range, $7 \leq Pe_{max} \leq 150$ shown in Fig. 6.8, and a value around 30.0 is used in this work. Through the Pe_{max} limit the estimated maximum free-surface velocity and the particle size, the damping viscosity can be calculated for each case.



(a)



(b)



(c)

Figure 6.7: The free-surface comparisons in dam break simulations, grey symbols are simulations without damping; black lines are free surfaces with $Pe_{max} = 30$ on the free surface. (a) $t \simeq 0.2s$; (b) $t \simeq 0.4s$; (c) $t \simeq 0.6s$.

To reduce the influence from free-surface damping, the viscosity value for particles adjacent to the free surface has been reduced to half or a quarter of free-surface viscosity.

It may be noted that a variable damping viscosity can also be calculated from the local velocity, particle size and a specified local Peclet number. The same dam break case is also simulated by specifying the local Peclet number of 10. The same effect as constant damping viscosity has been observed, shown in Fig. 6.9. In the results presented below a constant value has been used for simplicity.

In the numerical experiment, the water column height at the left side, H, and the water front edge, X, are recorded, comparing with experimental results from [57] and WCSPH results from [116] with the $k-\epsilon$ turbulence model. In Fig. 6.10, it is shown that both ISPH and WCSPH obtained quite similar results, both overpredict the water front although the column height predictions match the experiment very well. The free-surface damping did not strongly influence the predictions of the water column height and the water front edge. Lots of reasons may cause the overprediction of the water front. It should be pointed out that both ISPH and WCSPH suffer from the truncated kernel error, which may be an explanation for the difference between simulations and experiments.

The similar numerical scheme was also applied in [89, 6], where the artificial dissipation term was used to diffuse discontinuities with the smoothing scale similar to that resolved by the numerical method. In the previous ISPH free-surface simulations in [63], instabilities were not apparent but the $k-\varepsilon$ turbulence model was used which effectively increases viscosity above

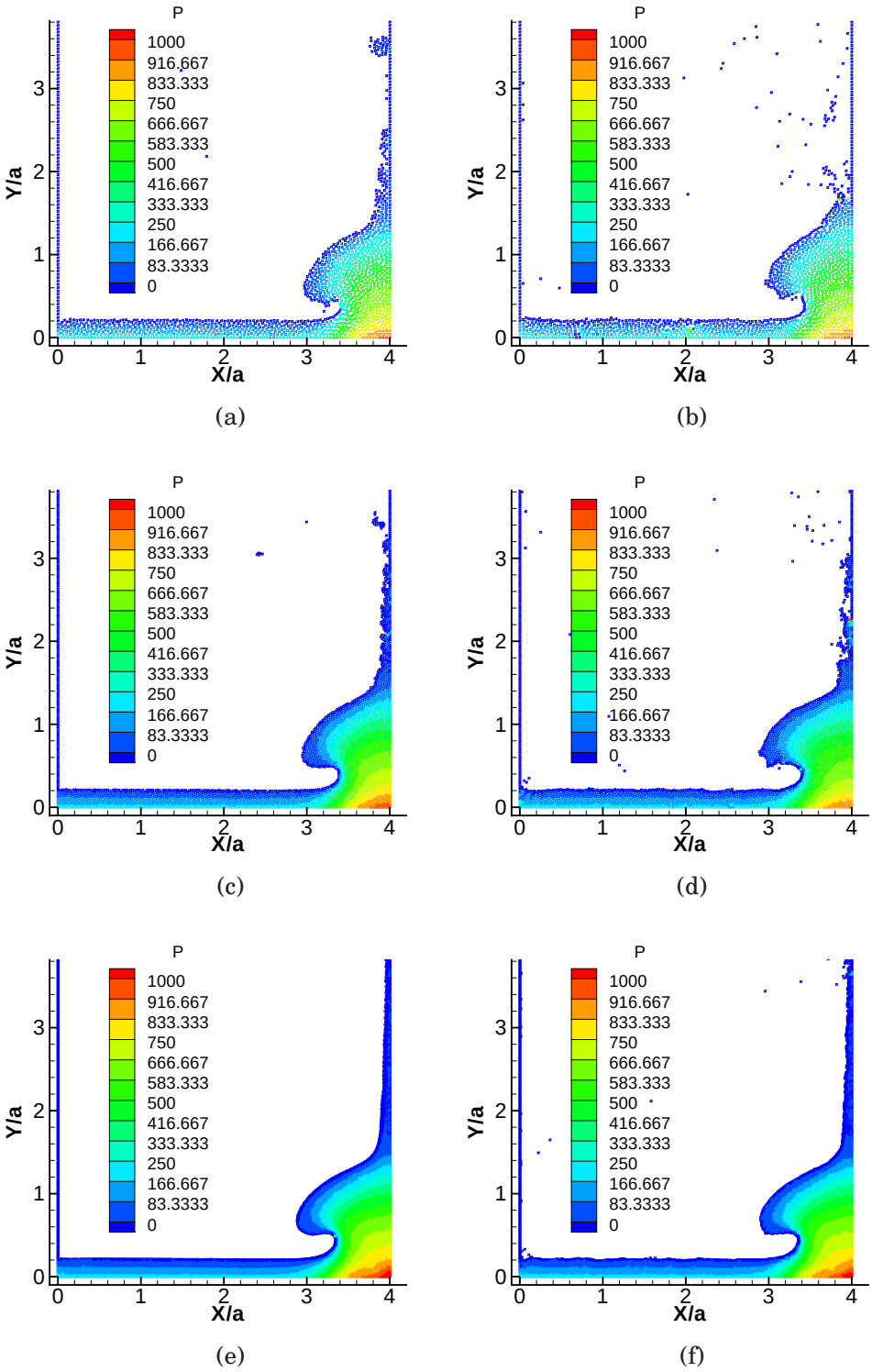


Figure 6.8: Graphs of the free-surface profiles and pressure contours in the collapse of water column, $t \simeq 0.6s$. (a) $Pe_{max} = 7.0$, $dx/a = 0.04$; (b) $Pe_{max} = 150.0$, $dx/a = 0.04$; (c) $Pe_{max} = 7.0$, $dx/a = 0.025$; (d) $Pe_{max} = 150.0$, $dx/a = 0.025$; (e) $Pe_{max} = 7.0$, $dx/a = 0.0125$; (f) $Pe_{max} = 150.0$, $dx/a = 0.0125$.

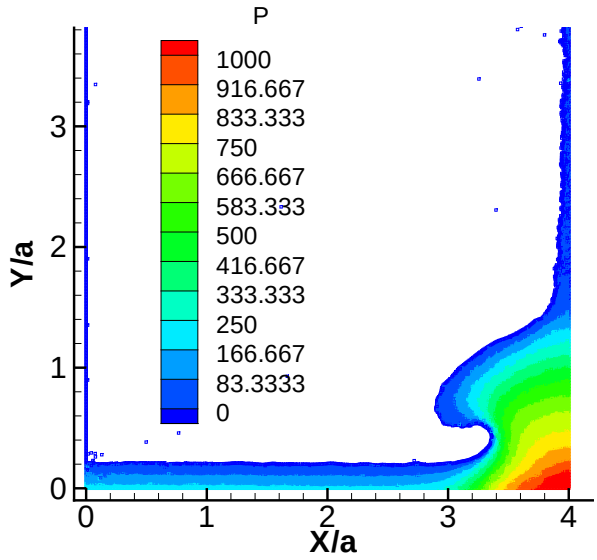


Figure 6.9: Graphs of the free surface and pressure contour in the collapse of water column, $t \simeq 0.6s$, $dx/a = 0.0125$. The damping viscosity is calculated from the local maximum Peclet number limit, which is 10 for free-surface particles, and 20 for particles close to the free surface.

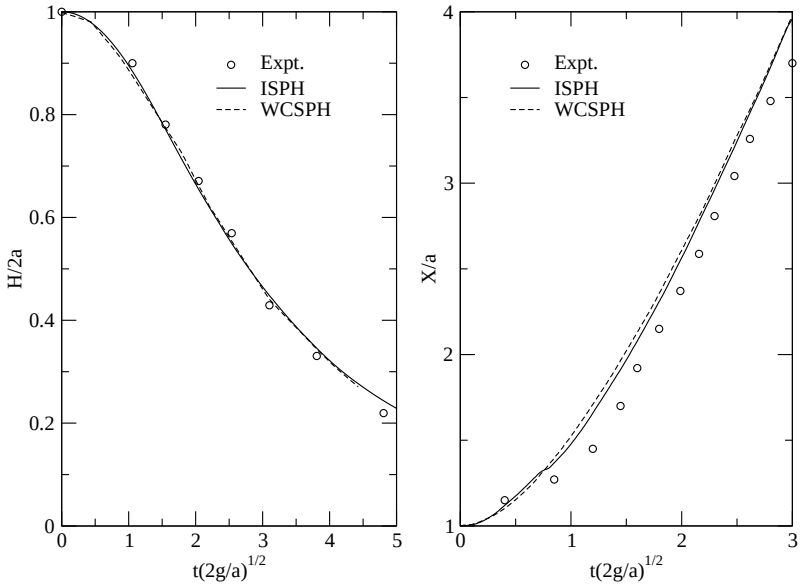


Figure 6.10: Collapse of a water column in a tank simulated. Non-dimensional column height $H/2a$ at the left side and water front position X/a. Comparison is made against the experimental data from [57] and WCSPH results from [116]. Solid line = ISPH results with a resolution of $12,\ 800$ fluid particles; Dashed line = WCSPH results from [116] with a resolution of $20,\ 000$ fluid particles; circles = experimental results from [57].

its molecular value. For almost inviscid fluids the effect on the accuracy of highly distorted flows and wave propagation needs to be assessed.

6.4 Simulation of impulsive paddle

6.4.1 Case description

A vertical plate in still water of depth $d = 0.5m$ is given an impulsive motion so that it moves instantaneously with steady velocity U. A jet of water travels up the face of the plate. The geometry is shown in Fig. 6.11.

An analytical solution for the free surface was derived by Peregrine [86] to first order in time, assuming zero gravity. The free-surface elevation, η, is written as

$$\eta = -\frac{2Ut}{\pi} ln\left(tanh\left(\frac{\pi x}{4h}\right)\right) \tag{6.2}$$

This transient free-surface flow has been further generalized by Roberts [95] for different paddle motions.

6.4.2 Simulation parameters

The free surfaces around the paddle, with two different paddle velocities, $U = 0.2m/s$ and $U = 1.0m/s$, are simulated for a water depth of $0.5m$. Two different initial particle spacings, $0.0125m$ and $0.00625m$, corresponding to 6380 and 25520 fluid particles, are used in both simulations. Due to the

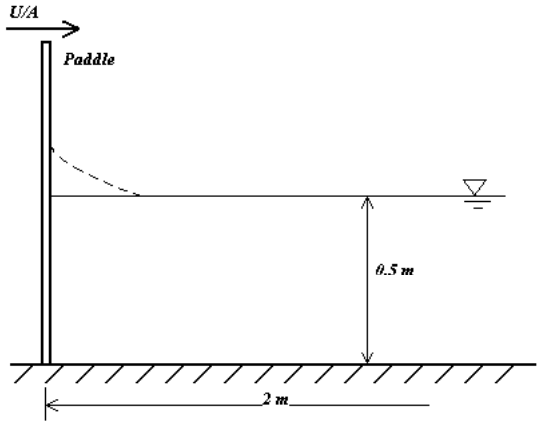


Figure 6.11: Impulsive paddle geometry

movement of the paddle, the number of solid particles on the bed is actually reduced according to the change of the bed length.

6.4.3 Simulation results

Fig. 6.12 shows the effect of Pe_{max} and Fig. 6.13 and 6.14 show the free-surface development with two different constant paddle velocities, $U = 0.2m/s$ and $U = 1.0m/s$ for different resolutions and $Pe_{max} = 30.0$. With smaller Pe_{max} limit, the free-surface predictions are less smooth than those with larger Pe_{max} limit, shown in Fig. 6.12. Generally, the ISPH predicts the free surface well with the limit $Pe_{max} = 30.0$. At the paddle, a jet is generated, but the free surface becomes scattered at the crest. At the crest, only very few particles are present in the SPH interpolation, therefore the truncated-kernel error will be higher here than elsewhere on the free surface. What is more, the value of $\nabla \cdot \mathrm{r}$ cannot accurately identify the free-surface particles, as shown in Fig. 6.13 and Fig. 6.14. Some particles inside of the free surface may also be taken as free-surface ones. The error caused by this

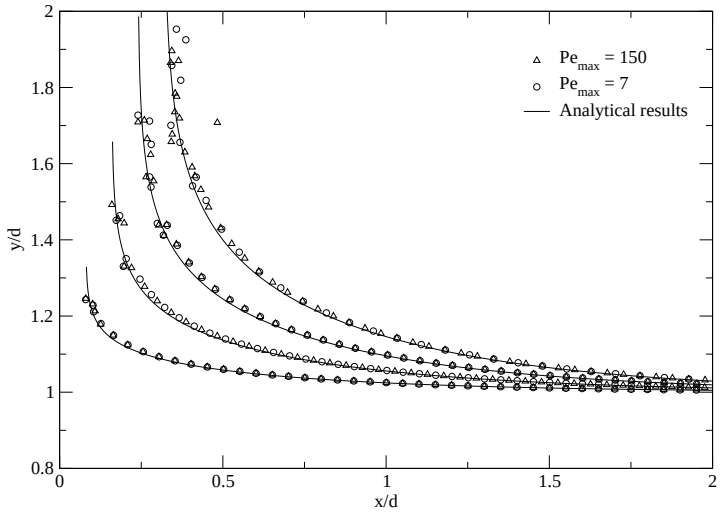


Figure 6.12: The free-surface simulations due to the impulsive paddle motion with different Pe_{max} limits, with a resolution of $dx = 0.0125$, a constant velocity $U = 1.0m/s$, at time $0.04s$, $0.08s$, $0.12s$, and $0.16s$, from left to right respectively.

mis-identification is also another reason for the scattered free surface at the crest of the jet.

In this simulation case, a higher resolution is also used in both cases shown in Figs. 6.13 and 6.14. More accurate predictions are obtained through the higher resolution simulations.

6.5 Simulation of 2-D dam break

6.5.1 Case description

The dam break is a widely used test case for impulsively started, rapid evolving free-surface flows. Experiments have been undertaken but results are affected by the withdrawal of the plate which has a similar time scale to the formation of the initial surface structures [102, 49]. For the dam break

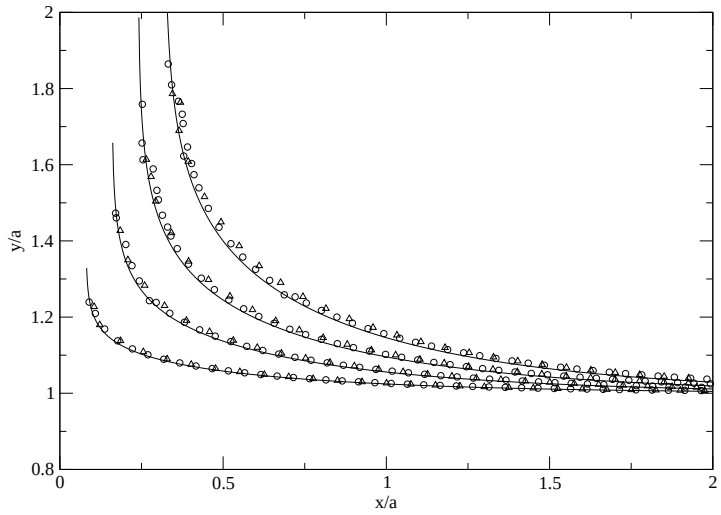


Figure 6.13: The free-surface simulations due to the impulsive paddle motion, with a constant velocity $U = 0.2m/s$, $Pe_{max} = 30.0$, at time $0.2s$, $0.4s$, $0.6s$ and $0.8s$, from left to right respectively. — = the analytical results [86]; $\triangle$ = the ISPH simulation with a resolution of $dx = 0.0125$; ○= the ISPH simulation with a resolution of $dx = 0.00625$.

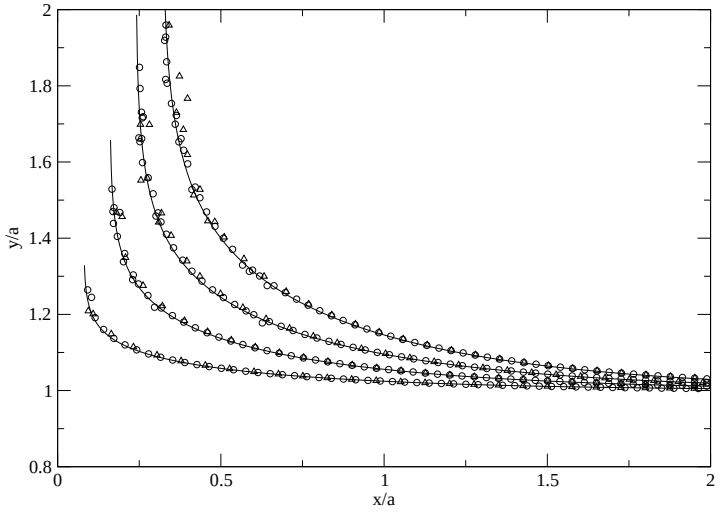


Figure 6.14: The free-surface simulations around the impulsive paddle, with a constant velocity $U = 1.0m/s$, $Pe_{max} = 30.0$, at time $0.04s$, $0.08s$, $0.12s$, and $0.16s$, from left to right respectively. — = the analytical results [86]; $\triangle$ = the ISPH simulation with a resolution of $dx = 0.0125$; ○= the ISPH simulation with a resolution of $dx = 0.00625$.

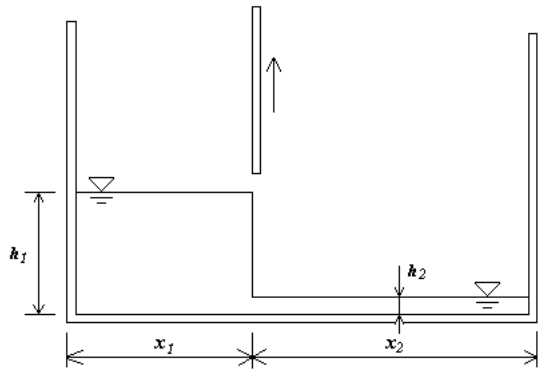


Figure 6.15: Geometry of dam break with wet bed. The board separating two water columns is removed or dissolved instantaneously.

with a wet bed, Stansby et al. [102] showed that the pressure distribution at $t = 0.0\,s$ and the resulting flow acceleration may be determined accurately by the solution of Laplace's equation for pressure. They also computed time evolution of the highly distorted free surface for small time assuming potential flow and solving by a highly accurate boundary integral method [24]. These solutions for $t = 0.0\,s$ and small times provide ideal tests for the ISPH algorithm developed here. The initial geometry is shown in Fig. 6.15. The gate separating the two columns is dissolved or removed immediately at $t = 0.0\,s$.

6.5.2 Simulation parameters

The fluid density ρ and kinematic viscosity ν are $1000kg/m^3$ and $1.0\times10^{-6}m^2/s$. In the investigation of the initial stage of dam break with the wet bed, the geometry parameters, h_1, h_2, x_1 and x_2, are set as $1.0m$, $0.1m$, $2m$ and $2m$ respectively. For $t = 0.0\,s$, the Laplacian equation for pressure is solved in finite difference form on a uniform mesh with different mesh sizes of $0.025m$, $0.02m$, $0.01m$, $0.005m$ and $0.0025m$, with the 2nd-order differencing. The lin-

ear system is solved by the successive over-relaxation (SOR) method. The same resolutions are used for ISPH simulations. The horizontal acceleration component a_x and the angle, θ, between the acceleration vector and the horizontal, from both methods are compared. The accelerations along the free-surface are plotted with results from both methods.

For the free-surface evolution within a small time, the free-surface profiles at different time are digitised from [102], and used as validation of ISPH predictions.

6.5.3 Simulation results

For the initial condition ($t = 0\,s$), Fig. 6.16 presents the pressure contours from both the Laplacian solution with the second-order finite difference method [102] and ISPH with a resolution of $0.01m$. The results are almost identical to each other. To compare further the results with each other, two cross-section pressure profiles, at $y = 0.0$ and $y = 0.5$, are plotted, as shown in Fig. 6.17. With a regular particle distribution, similar to a uniform structured mesh in finite difference or finite volume methods, the ISPH and Laplacian solvers provide an almost identical prediction.

Fig. 6.18 presents the acceleration profiles from both the ISPH and Laplacian solvers along the free surface, with Fig. 6.18 (a) showing horizontal acceleration values a_x at the free surface along $x = 2.0m$, and Fig. 6.18 (b) showing vertical acceleration values a_y at the free surface along $y = 0.1m$. The ISPH and Laplacian solvers provide almost identical predictions. ISPH however predicts slightly smaller values at the singularity point $x = 2.0m$ and

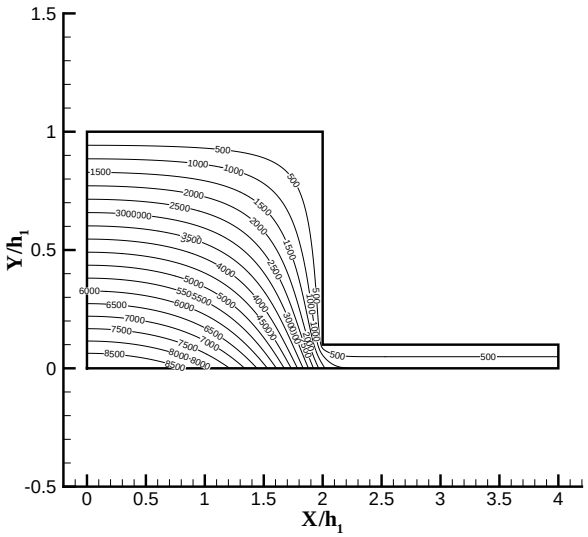
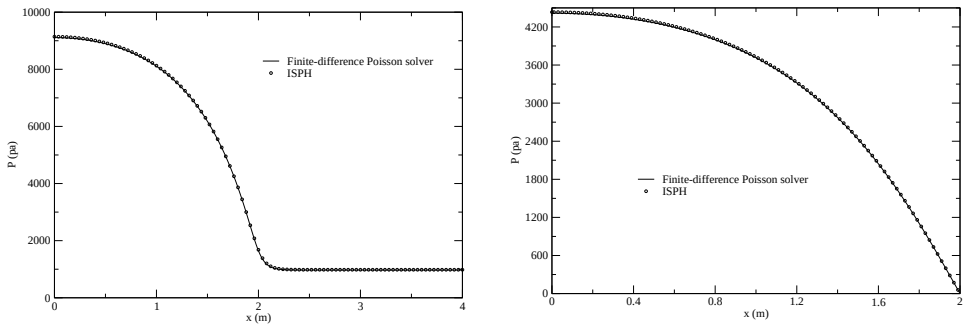
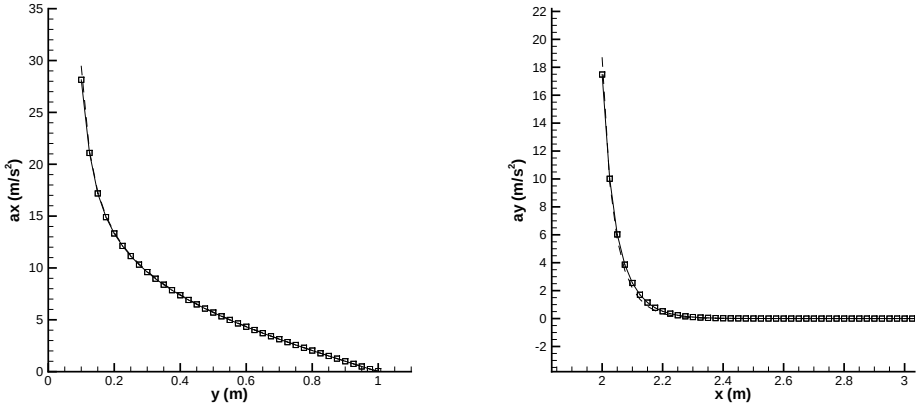


Figure 6.16: Comparison of the pressure P (Pa) between the finite-difference Laplacian solver and ISPH at $t = 0.0s$.$\cdots\cdots$: ISPH; — : finite-difference Laplacian solver.



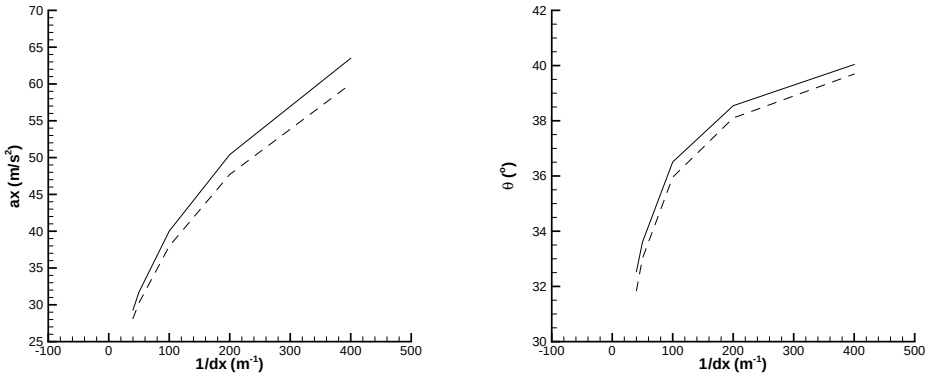
(a) (b)

Figure 6.17: Pressure profiles at two different cross sections, (a) $y = 0.0\,m$ and (b) $y = 0.5\,m$, at $t = 0.0s$.



(a) (b)

Figure 6.18: The fluid acceleration comparison along the free surface at $x = 2.0m$ and $y = 0.1m$ with the same initial mesh/particle size, $dx = dy = 0.025m$. Solid line with square symbols : ISPH; Dashed line : finite-difference Laplacian solver. (a) Profile of horizontal acceleration values a_x at the free surface $x = 2.0m$; (b) Profile of vertical acceleration values a_y at the free surface $y = 0.1m$.



(a) (b)

Figure 6.19: The fluid acceleration comparison at the singularity point $x = 2.0m$ and $y = 0.1m$. Solid line : finite-difference Laplacian solver; Dashed line : ISPH. (a) Profile of horizontal acceleration values a_x against resolutions $1/dx$; (b) Profile of angle θ, between the acceleration vector and the horizontal line, against resolution $1/dx$.

$y = 0.1m$ relative to the Laplacian solver. The behavior at the singularity is further investigated with different resolutions, shown in Fig. 6.19. In Fig.

6.19 (a), the horizontal accelerations from both methods are increasing with the increase of the resolution, and in Fig. 6.19 (b), the angle θ, between acceleration vector and the horizontal direction are converging towards about 45^o. The accelerations from ISPH are slighter lower but identical predictions at a singularity by different schemes are not expected, although they are expected in continuous regions.

For small time evolution, shown in Fig. 6.20 and 6.21, the predictions of the free surface from ISPH with $Pe_{max} = 7.0$, $Pe_{max} = 150.0$ are compared with digitised data from [102]. It is shown in Fig. 6.20 that ISPH with $Pe_{max} = 7.0$ damps some feature of the small-scale motion on the free surface, which is indicated by the loss of accurate mushroom-jet prediction, while in Fig. 6.21 ISPH with $Pe_{max} = 150.0$ provides small noise distorting the capture of the small-scale motion. By increasing the Pe_{max} limit to 30.0, the ISPH prediction generally matches the results based on the nonlinear potential-flow theory very well. However, at $t = 0.08s$ in Fig. 6.22, the cap of the mushroom-shape jet slightly deviates from the prediction in [102], which may be caused by the truncated-kernel error along the free surface. It should be noted that in [102] a tanh profile was used to define the initial condition to avoid sharp corners as required for the boundary integral method. This only changed the SPH distribution by one or two particles and had negligible influence on the results.

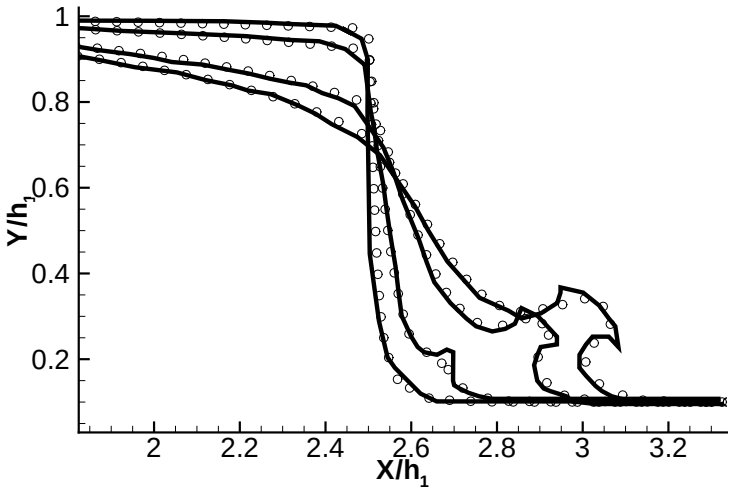


Figure 6.20: Free surface comparison at the initial stage of dam break with wet bed. From the left to the right, the solid lines and groups of symbols show the free surface at $t = 0.024s$, $t = 0.04s$, $t = 0.066s$ and $t = 0.08s$. — : digitised free-surface elevations from [102]; ∘ : free-surface particles from the ISPH simulation with $Pe_{max} = 7$.

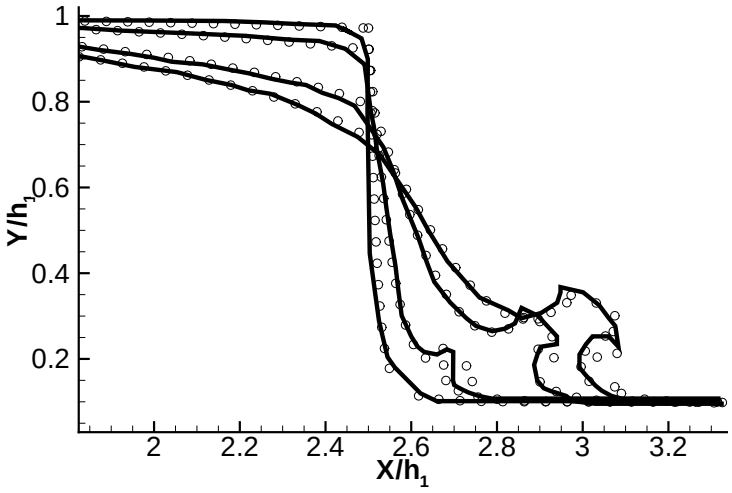


Figure 6.21: Free surface comparison at the initial stage of dam break with wet bed. From the left to the right, the solid lines and groups of symbols show the free surface at $t = 0.024s$, $t = 0.04s$, $t = 0.066s$ and $t = 0.08s$. — : digitised free-surface elevations from [102]; ∘ : free-surface particles from the ISPH simulation with $Pe_{max} = 150$.

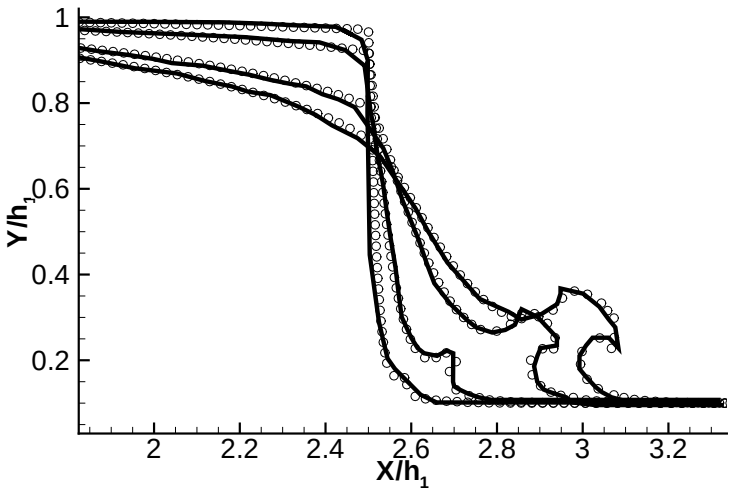


Figure 6.22: Free surface comparison at the initial stage of dam break with wet bed. From the left to the right, the four solid lines and groups of symbols show the free surface at $t = 0.024s$, $t = 0.04s$, $t = 0.066s$ and $t = 0.08s$. — : digitalised free-surface elevations from [102]; ∘ : free-surface particles from the ISPH simulation with $Pe_{max} = 30$.

6.6 Simulation of linear wave propagation along the channel

6.6.1 Case description

In this work, regular waves with wave heights, $H = 0.05m$ and $H = 0.1m$, in water of depth $d = 0.5m$ are simulated with the ISPH method. Waves are generated by the piston-type paddle in a rectangular wave tank with a length of $18m$. Based on linearized wavemaker theory [23], the paddle motion follows

$$u = \frac{S}{2}\omega cos(\omega t). \tag{6.3}$$

where u is the paddle velocity; S is the stroke of the wavemaker; ω is the wavemaker frequency. The stroke of the wavemaker S has the relation with

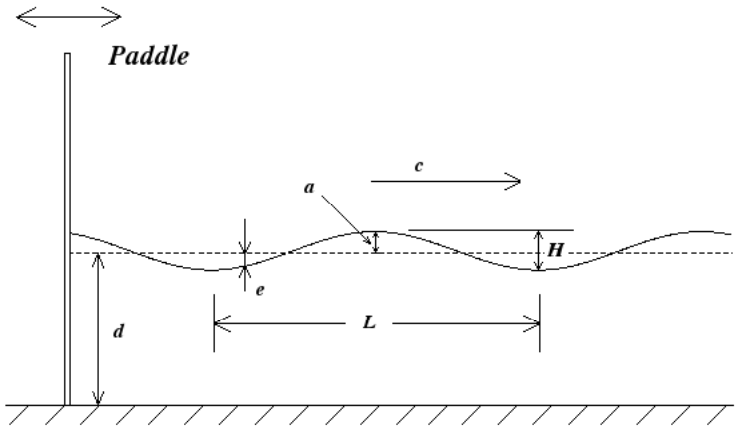


Figure 6.23: Waves generated by the paddle. a is the wave amplitude; H is the wave height; d is the water depth; L is the wave length; e is the free-surface elevation; c is the wave speed.

the wave height H, given by

$$\frac{H}{S} = \frac{2\left(cosh\, 2kd - 1\right)}{sinh\, 2kd + 2kd}, \tag{6.4}$$

and d is the water depth; k is the wave number.

$$k = 2\pi/l \tag{6.5}$$

The angular wave frequency ω in Eq. 6.3 has the relation with the wavenumber as

$$\omega^2 = gk \cdot tanh\ (kd)\,. \tag{6.6}$$

The wave parameters and geometry are illustrated in Fig. 6.23.

Even with small wave heights waves show slightly non-sinusoidal form and comparisons are made with highly accurate stream function theory of Rie-

Wave Heights H (m)	Wave Periods T (s)	Wave Length l (m)
0.05	0.9951	1.5
0.1	1.5688	3.0

Table 6.1: Parameters for different wave conditions

necker and Fenton [94] based on Fourier approximations, generalized in code SAWW (Steady Arbitrary Water Waves) by Buss and Stansby [8].

6.6.2 Simulation parameters

The wave conditions are listed in Table 6.1. The water depth d is $0.5\,m$, the wave tank length L is $18m$. The fluid density ρ is $1000kg/m^3$, the kinematic viscosity ν $10^{-6}m^2/s$, the initial particle spacing δr is $0.005\,m$.

6.6.3 Wave reflection absorption

To absorb the wave reflection from the wall at the end of the open channel, an exponential damping zone is placed over a distance of at least a wavelength. In the damping zone, the velocity of fluid particles will be damped as

$$\mathbf{U} = \mathbf{U}_0 f(x) \tag{6.7}$$

where U is fluid velocity, and

$$f(x) = 1 - e^{(-\alpha(\delta x_0 - (x - x_0)))}. \tag{6.8}$$

α is a coefficient, equal to 2.0; δx_0 is the damping zone length, equal to $3.0m$; x_0 is the damping zone starting point, equal to $15.0m$, $x_0 = L - \delta x_0$, and L is

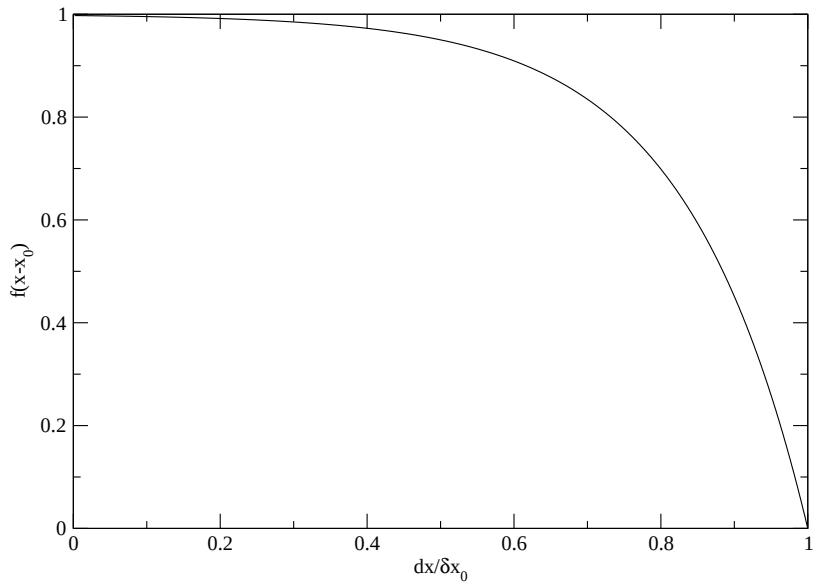


Figure 6.24: Profile of exponential damping function used for wave absorption. $\alpha = 2.0$; $\delta x_0 = 3.0$; $x_0 = 0.0$; $dx = x - x_0$.

the channel length. A similar exponential scheme has been used by Larsen and Dancy [61] for Boussinesq wave modelling. It is shown in Fig. 6.24 that when the particles are getting close to the right end of the wave tank, the velocity U will decrease to 0. This simple method effectively reduces the reflection of the wave from the right end.

6.6.4 Simulation results

Fig. 6.25 shows the wave propagation case with different damping viscosities. The wave with a wave height of $0.05m$ and a wave length of $1.5m$ is simulated. The wave tank length is set to $5m$ to reduce computing expense. The same Pe_{max} limits have been used as those in the previous numerical tests. It is shown that with $Pe_{max} = 150.0$, the free-surface prediction is not

as smooth as with $Pe_{max} = 7.0$. However, since it has been observed in earlier simulations that using too small Pe_{max} number limit can introduce too much numerical diffusion, $Pe_{max} = 30.0$ has again been chosen for the following wave propagation cases.

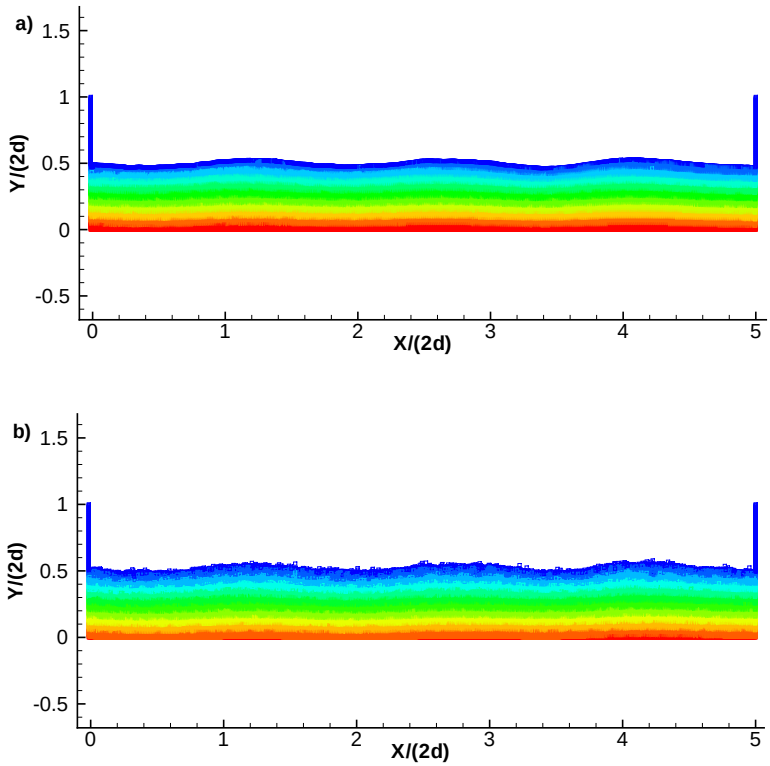


Figure 6.25: Damping investigation for wave propagation. (a) $Pe_{max} = 7$; (b) $Pe_{max} = 150$.

Fig. 6.26 and 6.28 show the wave propagation along the open channel for two different wave heights, $H = 0.05m$ and $H = 0.1m$, with snapshots of the free surface at different times. It is observed from the graphs that the first wave decayed soon after it is generated by the paddle. The second wave decayed gradually during the propagation process. But the other waves propagate along the channel without decaying. It can be observed that the damping zone effectively reduces the reflection from the right end of the channel under

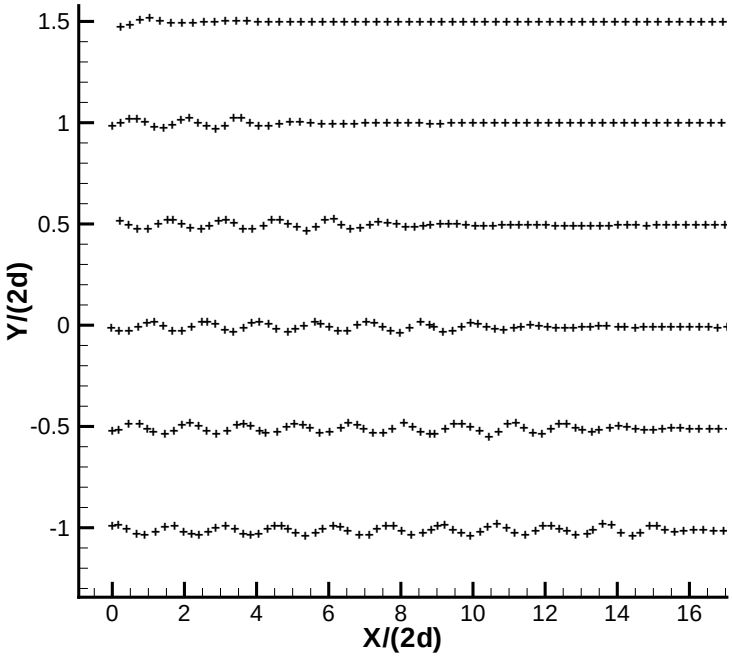


Figure 6.26: Wave propagation along the channel, from the top to the bottom, $t = 1.70625s$, $t = 5.3625s$, $t = 9.01875s$, $t = 12.675s$, $t = 16.33125s$, $t = 19.9875s$, the wave height $H = 0.05m$, the wave length $L = 1.5m$, the water depth $d = 0.5m$, $Pe_{max} = 30.0$. $+$: the free-surface particles in the simulation.

these situations, with very small reflection apparent in the last profiles in both Fig. 6.26 and 6.28.

Fig. 6.27 and 6.29 show comparisons of free-surface predictions from ISPH and stream function theory (code SAWW) at $t = 19.99s$ and $t = 9.75s$ for the two different wave heights, $H = 0.05m$ and $H = 0.1m$, respectively. It can be observed that for the larger wave height, waves are noticeably non-sinusoidal but a generally good agreement between ISPH and SAWW is obtained, with SAWW describing all nonlinear effects.

Fig. 6.30 and 6.31 present the total pressure, non-hydrostatic pressure and velocity profiles predicted by both ISPH and SAWW, below the crest and

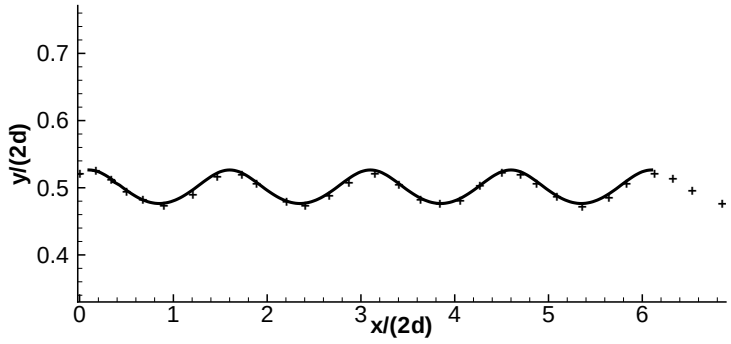


Figure 6.27: Comparison of free-surface predictions from both ISPH with $Pe_{max} = 30.0$ and SAWW [8]. Wave height $H = 0.05m$; Wave length $l = 1.5m$; Water depth $d = 0.5m$; $t = 19.9875s$.
$+$: free-surface prediction from ISPH; — : free-surface prediction from SAWW.

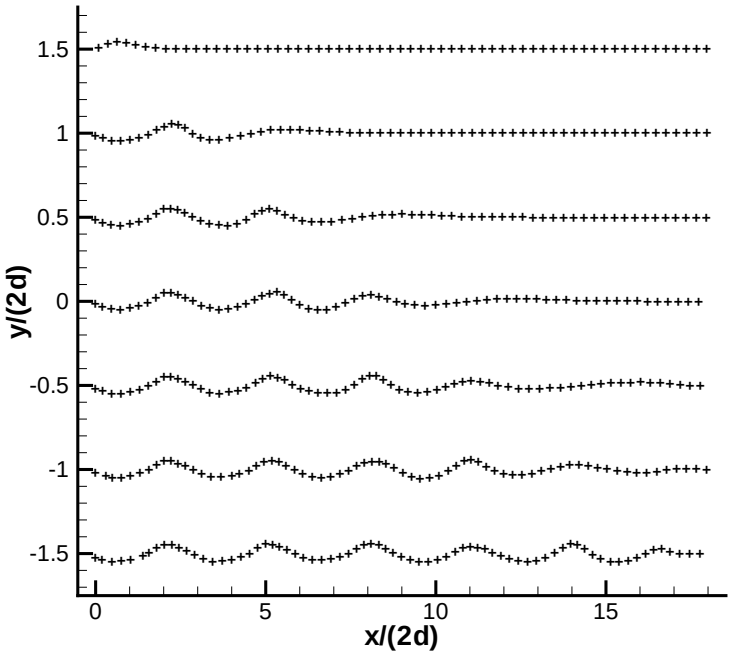


Figure 6.28: Wave propagation along the channel, from the top to the bottom, $t = 0.78s$, $t = 3.12s$, $t = 4.68s$, $t = 6.24s$, $t = 7.8s$, $t = 9.36s$, $t = 12.48s$, the wave height $H = 0.1m$, the wave length $L = 3.0m$, the water depth $d = 0.5m$, $Pe_{max} = 30.0$. $+$: the free-surface particles in the simulation.

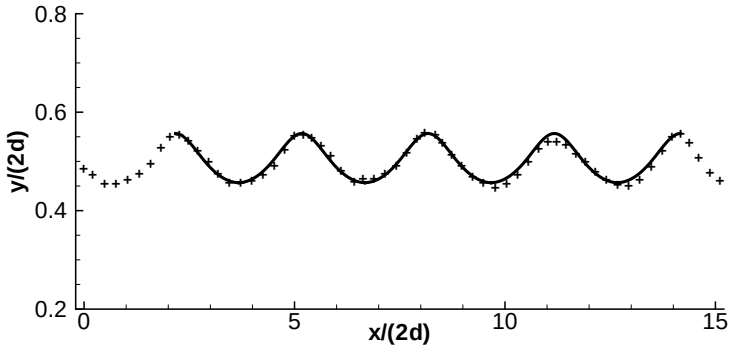


Figure 6.29: Comparison of free-surface predictions from both ISPH with $Pe_{max} = 30.0$ and SAWW [8]. Wave height $H = 0.1m$; Wave length $l = 3.0m$; Water depth $d = 0.5m$; $t = 9.75s$.
$+$: free-surface prediction from ISPH; — : free-surface prediction from SAWW.

trough, with positions given on the figure caption. The non-hydrostatic pressure is given by

$$p = P - \rho g(\eta - y), \tag{6.9}$$

where p is the non-hydrostatic pressure; P is the total pressure calculated from ISPH; η is the free-surface elevation. It is shown that ISPH method predicts the fluid field accurately for both the velocity and total pressure. The non-hydrostatic pressure predictions from ISPH are however slightly different from the predictions from code SAWW. The reason for this is uncertain. It could be due to the effect of the wave maker being apparent in comparison with an infinite wave train. The matter remains to be resolved.

6.7 Partial conclusion

An ISPH algorithm has been applied to almost inviscid free-surface flows. It has been observed that instability can develop at the free surface due to

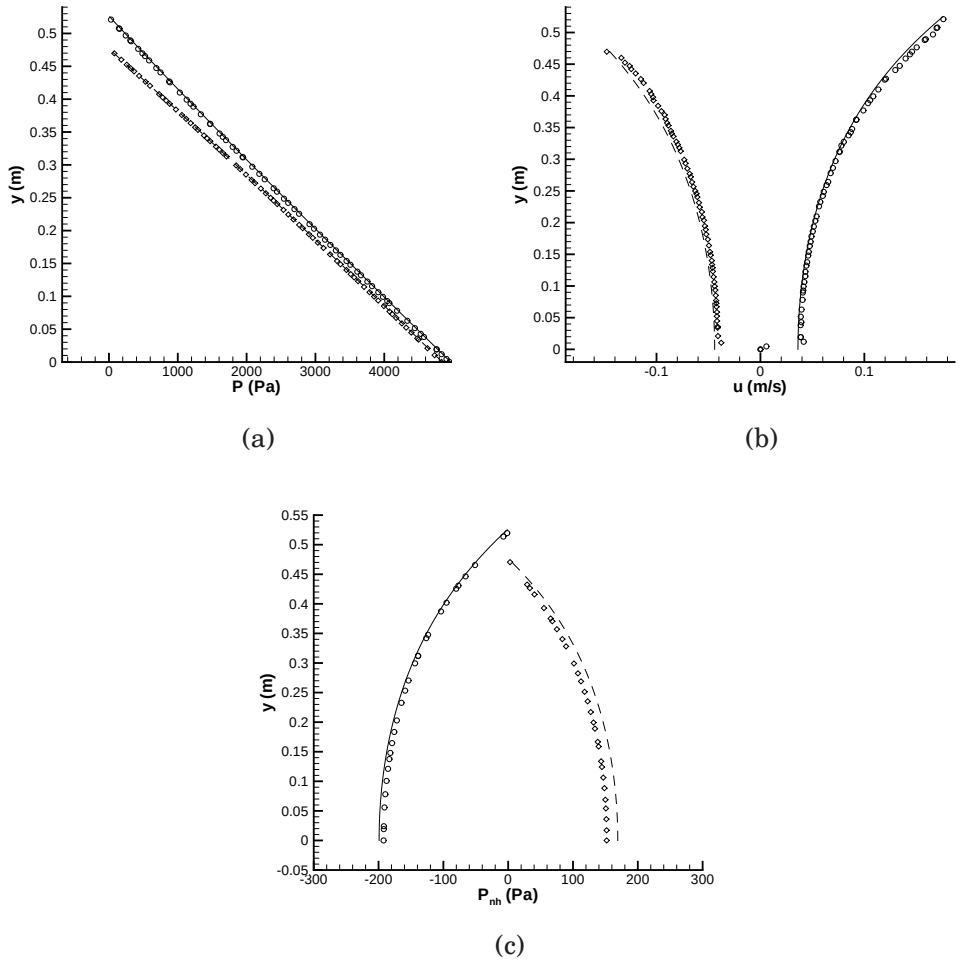
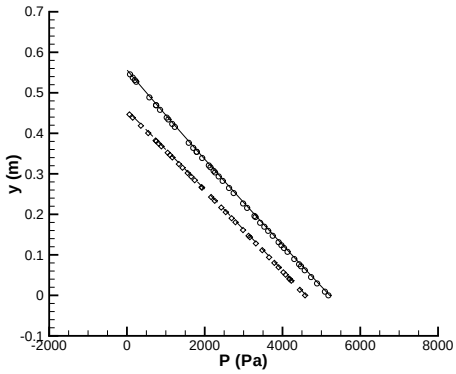
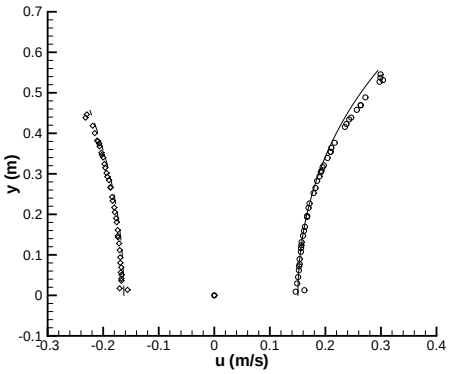


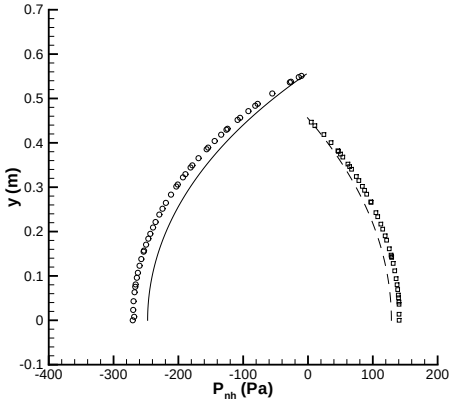
Figure 6.30: Comparing velocity and pressure predictions from both ISPH with $Pe_{max} = 30.0$ and SAWW [8].Wave height $H = 0.05m$; Wave length $l = 1.5m$; Water depth $d = 0.5m$; $t = 19.9875s$.
(a) Pressure profiles below wave crest and trough; (b) Horizontal velocity profiles below wave crest and trough; (c) Non-hydrostatic pressure profiles below wave crest and trough.
— : prediction from SAWW below the wave crest ; $-\ -\ -$: prediction from SAWW below the wave trough ; $\circ$: prediction from ISPH along below the wave crest, $x = 1.61m$; $\diamond$: prediction from ISPH line below the wave trough, $x = 2.36m$.



(a)



(b)



(c)

Figure 6.31: Comparing velocity and pressure predictions from both ISPH with $Pe_{max} = 30.0$ and SAWW [8].Wave height $H = 0.1m$; Wave length $L = 3.0m$; Water depth $d = 0.5m$; $t = 9.75s$
(a) Pressure profiles below wave crest and trough; (b) Horizontal velocity profiles below wave crest and trough; (c) Non-hydrostatic pressure profiles below wave crest and trough.
— : prediction from SAWW below the wave crest ; $-\ -\ -$: prediction from SAWW below the wave trough ; $\circ$: prediction from ISPH below the wave crest, $x = 5.943m$; $\diamond$: prediction from ISPH below the wave trough, $x = 7.443m$.

the truncated kernel. This may be reduced by increasing kinematic viscosity at the surface and, with sufficient particle resolution and surface Pe_{max} numbers below about 30, accurate flow predictions can be made. Surface Pe_{max} numbers may also be reduced by increasing particle resolution without changing viscosity but this would increase computational times considerably. The algorithm with artificially increased surface viscosity is tested for the impulsive flows of the impulsively started plate and the wet bed dam break for which there are analytical and highly accurate solutions. It is also shown that regular wave propagation can be accurately simulated without decay. Surface profiles, total pressures and velocities are simulated accurately, although apparent small difference in non-hydrostatic pressure remain to be resolved.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, the stability and accuracy of existing incompressible SPH methods, ISPH_DF [17, 63], ISPH_DI [100], ISPH_DFDI [44] have been investigated. Success has been achieved in several different applications. However, it has been discussed that due to the kernel flaw and the streamline movement of particles in the SPH method , ISPH_DF [17, 63] is not a stable method; the simulations with ISPH_DI [100] are quite stable, but the inaccurate prediction has dropped the advantage to use incompressible SPH; ISPH_DFDI [44] can provide accurate predictions but with much higher computing expense and doubled pressure and pressure gradient fields.

Based on the ISPH_DF [17, 63], the ISPH_DFS is a stable algorithm without sacrificing efficiency. Particles are shifted slightly across streamlines, avoiding particle spacing distortion and the error resulting from particle clustering, and their hydrodynamic properties are adjusted by Taylor series inter-

polation. This small change to ISPH_DF method has stabilized the ISPH method with very little increase of computing expense. Through the test cases, Taylor-Green vortices, vortex spindown, it has been shown that that ISPH_DFS method is much more efficient than ISPH_DFDI method [44]; the accuracy is improved over ISPH_DI method [100]; the stability is highly enhanced comparing with ISPH_DF method [17, 62].

The accuracy of ISPH_DFS method has been studied in Chapter 5. Through the error estimation in Taylor-Green vortices simulations the spatial convergence speed of ISPH_DFS method has been shown to be first order. As the gradient and divergence operators have already been shown to be second order [82], the influence of the viscosity and Laplacian operators is investigated. It is shown in §5.1 that although the operator proposed by Schwaiger [99] has improved the accuracy performance, the spatial convergence speed is not enhanced. Similar convergence speeds have been obtained for both types of the viscous and Laplacian operators [77, 99]. This is most possibly caused by the particle convection method – particles move after the velocity at the new time step is updated, but will have the same velocity at the new place as at the old position. This delayed velocity updating may be one of main reasons causing slow spatial convergence.

Two benchmark cases, lid-driven cavity and bluff body in the channel, were simulated. The results are compared with Ghia's data [32] and the finite volume simulations from the commercial software STAR-CD and code Saturne [1]. It has been shown that for low Reynolds number, ISPH method gave quite good prediction as finite volume method did, while the low convergence speed limits the accomplishment of acceptable results with low resolutions, which is consistent with the first-order spatial convergence speed. The drag

and lift forces were calculated in the bluff-body cases. It has been shown that the movement of the particles around the circular cylinder has resulted in the noisy force values. This noise may disable or place difficulty on the prediction of the vortex-shedding frequency under the situation of low resolutions. Although the increase of the resolution can provide higher accuracy without considering the computing expense, a better approach for the simulation of fluid on the wall is needed, especially for the accurate simulation of fluid-body interaction.

In this thesis, the influence of the truncated-kernel error to the free-surface prediction has also been discussed. With corrected kernels for different operators, the error on the free surface can be highly reduced for uniformly distributed particles, while this small error will be amplified as the particle distribution is distorted, and the free surface could be quite noisy with ISPH method. One treatment has been introduced here by slightly increasing the viscosity on the free surface. This has been proved to be an effective way to reduce the noise on the free surface without strongly influencing the free-surface prediction.

Three test cases, initial-stage investigation to the dam break, impulsive paddle and wave propagation, have been studied. Good agreements are obtained for the studied cases. The thorough quantitative validation has been conducted for pressure and velocity predictions in the context of the SPH method through the wave propagation test case. It is shown that for the free-surface prediction, or for the velocity and pressure field, this ISPH method can provide accurate predictions.

7.2 Future work and suggestions

In this work, a new ISPH algorithm to avoid instability caused by highly-distorted particle distributions is proposed. The accuracy and stability are well maintained without sacrificing the efficiency. However, it is shown that this ISPH method only converges with first-order behavior as the spatial resolution increases. The low-accuracy operators and time-marching scheme may be the cause of the low-order convergence speed. Although the gradient or divergence operators has been proved to be second-order [82], the Laplacian or viscous operators are not. Higher order interpolation methods are needed. In the existing algorithm, particles move explicitly with Crank-Nicolson scheme. In the new position, particles are still having the old hydrodynamic variable values, such as velocity. Therefore, an investigation of fully-implicit time marching is necessary.

In this work, all the simulated cases are with simple geometries. In this way, the complexity caused by the geometries is avoided, and the boundary condition problem is bypassed. In ISPH, the non-slip and numerical Neumann boundary conditions are needed for velocity and pressure respectively. The existing methods, mirror particle and dummy particle as presented in this thesis, still have difficulties to build accurate boundary conditions for complicated geometries.

At the end, the truncated-kernel error around free surface has been investigated in this work. A numerical approach to overcome the instability caused by the truncated-kernel error on the free surface has been proposed. However, it should admitted here that this approach is mainly a numerical method. A better way is needed.

Appendix A

The SPHysics Code

Here, only a brief instruction about code SPHysics will be presented to facilitate the further illustration about the incompressible SPH code, which is developed based on this WCSPH code SPHsysics. More details about SPHysics can be obtained from the user guide [121].

Smoothing kernel

The accuracy of SPH simulations is strongly influenced by the choice of kernel function. The kernel function is actually a mathematical approximation to the Dirac δ function. The interesting ones for our purpose have a compact support area. The smoothing function ω_{ij} should monotonically decrease as the distance between particle i and its neighbouring particle j increase. The smoothing length h is proportional to the initial particle distance dr, normally $h = c \cdot dr$, and decides the support size of the particle i, L_i. For different kernels in a manner of linear algebraic equations, $L_i = \alpha h$ with different coefficient α value. The non-dimensional distance q is used in the kernel calculation, and $q = r_{ij}/h \geqslant 0$, where r_{ij} is the distance between particle i and j.

In SPHysics, four different kernel functions are available: Gaussian kernel, quadratic kernel, cubic kernel and quintic kernel.

Gaussian kernel [73]

$$\omega(q) = \gamma e^{\left(-q^2\right)} \tag{A.1}$$

where, γ is a coefficient, equal to $1/\left(\pi h^2\right)$ in 2D, and $1/\left(\pi^{3/2}h^3\right)$ in 3D.

Quadratic kernel [18]

$$\omega\left(q\right) = \gamma \begin{cases} \left(\dfrac{3}{16}q^2 - \dfrac{3}{4}q + \dfrac{3}{4}\right) & 0 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases} \tag{A.2}$$

where, γ is equal to $2/\left(\pi h^2\right)$ in 2D, and $5/\left(4\pi h^3\right)$.

Cubic kernel [70]

$$\omega\left(q\right) = \gamma \begin{cases} 1 - \dfrac{3}{2}q^2 + \dfrac{3}{4}q^3 & 0 \leq q \leq 1 \\ \dfrac{1}{4}\left(2-q\right)^3 & 1 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases} \tag{A.3}$$

where, γ is equal to $10/\left(7\pi h^2\right)$ in 2D, and $1/\left(\pi h^3\right)$ in 3D.

Quintic kernel [119]

$$\omega\left(q\right) = \gamma \begin{cases} \left(1 - \dfrac{q}{2}\right)^4 \left(2q+1\right) & 0 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases} \tag{A.4}$$

where, γ is equal to $7/\left(4\pi h^2\right)$ in 2D, and $7/\left(8\pi h^3\right)$ in 3D.

Momentum equation

Artificial viscosity and Sub-Particle Scale model

In the momentum equation Eq. 2.2, one strong effect is viscosity. The viscous effect can be simulated by the artificial viscosity [71] or by the approximated laminar viscosity [77]. Excluding the artificial viscosity and the laminar viscosity, the Sub-Particle Scale (SPS) approach, which was first introduced by Gotoh et al. [34] in the context of Moving Particle Semi-implicit method (MPS), to model turbulence was also developed in the code SPHysics.

The artificial viscosity was first proposed by Monaghan [71]. Due to its simplicity, it is widely used in the SPH simulations. If we define the viscous term in momentum equation as ψ_i, the formulation reads

$$\psi_i = \sum_j m_j \Pi_{ij} \nabla \omega_{ij} \tag{A.5}$$

$$\Pi_{ij} = \begin{cases} \dfrac{-\zeta \overline{c}_{ij} \mu_{ij}}{\overline{\rho}_{ij}} & \mathbf{u}_{ij} \cdot \mathbf{r}_{\mathrm{ij}} < 0 \\ 0 & \mathbf{u}_{ij} \cdot \mathbf{r}_{\mathrm{ij}} > 0 \end{cases} \tag{A.6}$$

and

$$\mu_{ij} = \frac{h \mathbf{u}_{ij} \cdot \mathbf{r}_{\mathrm{ij}}}{r_{ij}^2 + \eta^2} \tag{A.7}$$

where, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$; $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$; $\overline{c}_{ij}$ is the averaged speed of sound for particle i and j, $\overline{c}_{ij} = \frac{c_i + c_j}{2}$; $\overline{\rho}_{ij}$ is the averaged density for particle i and j, $\overline{\rho}_{ij} = \frac{\rho_i + \rho_j}{2}$; $\eta = 0.1h$; ζ is a coefficient, changed depending on the simulations.

SPS model, similar to Large Eddy Simulation, is used to consider the effect of fluid scale smaller than the particle size and simulate the turbulence effect in the fluid, seeing Gotoh et al. [34]. In 2004, Gotoh et al. [35] extended the SPS model into incompressible SPH. To include the sub-particle effect in the fluid, the momentum equation is changed into

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + F + \frac{1}{\rho}\nabla\tau \tag{A.8}$$

where F is the body force; τ is the SPS stress tensor. With the Boussinesq's hypothesis, SPS stress tensor is modeled as

$$\tau_{ij} = \rho\left(2\nu_t S_{ij} - \frac{2}{3}k\delta_{ij} - \frac{2}{3}C_I\Delta^2\delta_{ij} \mid S_{ij} \mid^2\right) \tag{A.9}$$

where, ν_t is the eddy viscosity, $\nu_t = \left[min\left(C_s\triangle l^2\right)\right]^2 \mid \mathbf{S} \mid$, $C_I = 0.0066$, S_{ij} is the stress tensor element, C_s is the Smagorinsky constant, with a range of 0.1 - 0.2, and a value of 0.12 is used in SPHysics code. $\triangle l$ is the particle-particle spacing, and it is used as the implicit filter here, equivalent to the grid size in mesh-based method; $\mid \mathbf{S} \mid = \left(2S_{ij}S_{ij}\right)^{1/2}$.

Pressure calculation and state equation

In original SPHysics code, the incompressible flow is assumed with weak compressibility. The pressure field is not solved from a pressure Poisson equation, but from a state equation. In the state equation, the speed of sound, whose value should be at least ten times of the maximum fluid velocity, is used for the calculation of coefficient. To satisfy the CFL condition, the time step is limited to a very small value, which slows down the simulation efficiency. Compressibility causes problems with sound wave reflection

at the boundaries. However, due to its simplicity, the state equation is still widely used in pressure calculation in the simulations. The state equation follows the expression

$$P = B\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{A.10}$$

where, $\gamma = 7$, $B = c_0^2\rho_0/\gamma$, with ρ_0 being the reference density and c_0 being the speed of sound at the reference density and $c_0 = c\left(\rho_0\right) = \sqrt{\frac{\partial P}{\partial \rho}}$.

Discretized momentum equation

With substituting the pressure gradient term and SPS stress gradient term with Eq. 2.14, the viscous term with Eq. 2.19, and considering the SPS model, the momentum equation, Eq. A.8 can be discretized as

$$\begin{aligned}\left(\frac{d\mathbf{u}}{dt}\right)_i \simeq & -\sum_j m_j\left(\frac{P_i}{\rho_i^2}+\frac{P_j}{\rho_j^2}\right)\nabla\omega_{ij} + \sum_j \frac{4m_j\mu\mathbf{r}_{ij}\cdot\nabla\omega_{ij}}{\left(\rho_i+\rho_j\right)\left(r_{ij}^2+\eta^2\right)}\mathbf{u}_{ij} \\ + & \ \mathbf{F} + \sum_j m_j\left(\frac{\tau_i}{\rho_i^2}+\frac{\tau_j}{\rho_j^2}\right)\nabla\omega_{ij}\end{aligned} \tag{A.11}$$

Continuity equation

For the compressible flow, the continuity equation is

$$\frac{d\rho}{dt} + \rho\nabla\cdot\mathbf{u} = 0. \tag{A.12}$$

Therefore, the particle density can be updated with Eq. 2.15 as the formulation

$$\left(\frac{d\rho}{dt}\right)_i = -\rho\nabla\cdot\mathbf{u} = \sum_j m_j\mathbf{u}_{ij}\cdot\nabla\omega_{ij}. \tag{A.13}$$

The right hand side of Eq. A.13 is also briefly written as D in the later part.

Also the particle density can be calculated from Eq. 2.6, which is

$$\rho = \sum_j m_j \omega_{ij}. \tag{A.14}$$

For the accuracy consideration, both of the expression suffer from the truncated kernel error at the free surface. For the efficiency consideration, Eq. A.13 needs the first derivative of the kernel function, but Eq. 2.41 the kernel value, which means extra memory and computing cost will be put in because the momentum equation needs only the first derivative.

Time marching

In code SPHysics, several different time marching schemes, Predictor-Corrector (also known as the modified Euler method) [72], Verlet and Beeman [3], are used. The Predictor-Corrector and Verlet scheme are introduced here. Others can be found in [31].

Predictor-Corrector

In Predictor-Corrector time marching scheme [72], the variable fields are first predicted with the half step $n + 1/2$ based on time step n, and then corrected based on the previous predicted stage, $n + 1/2$.

Predicting stage:

$$\mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n + \frac{\Delta t}{2}\mathbf{F}^n \tag{A.15}$$

$$\rho_i^{n+1/2} = \rho_i^n + \frac{\Delta t}{2}D^n \tag{A.16}$$

$$\mathbf{r}_i^{n+1/2} = \mathbf{r}_i^n + \frac{\Delta t}{2}\mathbf{u}^n \tag{A.17}$$

Correcting stage:

$$\mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n + \frac{\Delta t}{2}\mathbf{F}^{n+1/2} \tag{A.18}$$

$$\rho_i^{n+1/2} = \rho_i^n + \frac{\Delta t}{2}D^{n+1/2} \tag{A.19}$$

$$\mathbf{r}_i^{n+1/2} = \mathbf{r}_i^n + \frac{\Delta t}{2}\mathbf{u}^{n+1/2} \tag{A.20}$$

Variable field at time $n+1$:

$$\mathbf{u}_i^{n+1} = 2\mathbf{u}_i^{n+1/2} - \mathbf{u}_i^n \tag{A.21}$$

$$\rho_i^{n+1/2} = 2\rho_i^{n+1/2} - \rho_i^n \tag{A.22}$$

$$\mathbf{r}_i^{n+1/2} = 2\mathbf{r}_i^{n+1/2} - \mathbf{r}_i^n \tag{A.23}$$

Verlet Scheme

In Verlet scheme, a correction step . The variables are updated as

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^{n-1} + 2\Delta t\mathbf{F}^n; \tag{A.24}$$

$$\rho_i^{n+1} = \rho_i^{n-1} + 2\Delta t D^n; \tag{A.25}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t\mathbf{u}_i^n + \frac{\Delta t^2}{2}\mathbf{F}^n. \tag{A.26}$$

To stop the integration diverging, every M time steps ($M = 50$ in SPHysics) variables are calculated according to

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta t\mathbf{F}^n; \tag{A.27}$$

$$\rho_i^{n+1} = \rho_i^n + \Delta t D^n; \tag{A.28}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t \mathbf{u}_i^n + \frac{\Delta t^2}{2}\mathbf{F}^n. \tag{A.29}$$

Appendix B

Linear Solvers

Conjugate gradient (CG) solver

CG is the most popular iterative solver. It is widely used to solve the Pressure Poisson equation in incompressible simulations. The CG method is an algorithm for the numerical solution of particular systems of linear equations, namely those whose matrix is symmetric and positive-definite. Therefore, the diffusion dominant problem, such as Pressure Poisson equations, will be the best target for this solver. The CG method is an iterative method, so it can be applied to sparse systems that are too large to be handled by direct methods such as the Cholesky decomposition. Such systems arise regularly when numerically solving partial differential equations. However, the abnormal behavior of the residual norm hinders the convergence of the iteration, even introduce instability in the simulations. For a linear system, $\mathbf{AX} = \mathbf{B}$, with an initial guessed value $\mathbf{X}_0$, the CG algorithm is listed below.

$$\mathbf{r}_0 = \mathbf{B} - \mathbf{A}\mathbf{X}_0 \tag{B.1}$$

$$\mathbf{P}_0 = \mathbf{r}_0 \tag{B.2}$$

Iterate the following conjugate-gradient root search process until the convergence criterion is satisfied,

$$\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{P}_k^T \mathbf{A} \mathbf{P}_k} \tag{B.3}$$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha_k \mathbf{P}_k \tag{B.4}$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{P}_k \tag{B.5}$$

And $k = 0$ at the first iteration. If the convergence criterion is satisfied, the iteration will be stopped. The residual vector may direct the iteration to divergence. Therefore, after several iterations, a correction step for residual will be performed.

$$\mathbf{r}_{k+1} = \mathbf{B} - \mathbf{A} \mathbf{X}_{k+1} \tag{B.6}$$

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \tag{B.7}$$

$$\mathbf{P}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{P}_k \tag{B.8}$$

$$k = k + 1 \tag{B.9}$$

Bi-Conjugate Gradient Stabilized (Bi-CGSTAB) solver

The Bi-CGSTAB algorithm with a preconditioner, a Jacobi preconditioner in the code, is simply listed below. The same as that in CG solver, the linear system is simply written as $\mathbf{AX} = \mathbf{B}$ with an initial guessed value $\mathbf{X}_0$ and a preconditioner $\mathbf{K}$, and the residual is

$$\mathbf{r}_0 = \mathbf{B} - \mathbf{A} \mathbf{X}_0 \tag{B.10}$$

$\bar{\mathbf{r}}_0$ is an arbitrary vector, and the dot product $(\mathbf{r}_0, \bar{\mathbf{r}}_0)$ satisfies

$$(\mathbf{r}_0, \bar{\mathbf{r}}_0) \neq 0 \; e.g. \; \bar{\mathbf{r}}_0 = \mathbf{r}_0. \tag{B.11}$$

The initial parameter setting can be performed as

$$\rho_0 = \alpha = \omega_0 = 1; \tag{B.12}$$

$$\mathbf{v}_0 = \mathbf{p}_0 = 0. \tag{B.13}$$

Then the following conjugated-gradient root searching process with the residual smoothing can be conducted until the convergence of the linear system is achieved.

$$\rho_i = (\bar{\mathbf{r}}_0, \mathbf{r}_{i-1}) \, ; \; \beta = (\rho_i / \rho_{i-1}) \, (\alpha / \omega_{i-1}) \, ; \tag{B.14}$$

$$\mathbf{p}_i = \mathbf{r}_{i-1} + \beta \, (\mathbf{p}_{i-1} - \omega_{i-1} \mathbf{v}_{i-1}) \, ; \tag{B.15}$$

$$Solve \; \mathbf{y} \; from \; \mathbf{K}\mathbf{y} = \mathbf{p}_i; \tag{B.16}$$

$$\mathbf{v}_i = \mathbf{A}\mathbf{y}; \tag{B.17}$$

$$\alpha = \rho_i / \left(\bar{\mathbf{r}}_0, \mathbf{v}_i\right) ; \tag{B.18}$$

$$\mathbf{s} = \mathbf{r}_{i-1} - \alpha \mathbf{v}_i; \tag{B.19}$$

$$Solve \; \mathbf{z} \; from \; \mathbf{K}\mathbf{z} = \mathbf{s}; \tag{B.20}$$

$$\mathbf{t} = \mathbf{A}\mathbf{z}; \tag{B.21}$$

$$\omega_i = \left(\mathbf{K}^{-1}\mathbf{t}, \mathbf{K}^{-1}\mathbf{s}\right) / \left(\mathbf{K}^{-1}\mathbf{t}, \mathbf{K}^{-1}\mathbf{t}\right) ; \tag{B.22}$$

$$\mathbf{X}_i = \mathbf{X}_{i-1} + \alpha \mathbf{y} + \omega_i \mathbf{z}. \tag{B.23}$$

If $\mathbf{X}_i$ is accurate enough then quite, otherwise

$$\mathbf{r}_i = \mathbf{s} - \omega_i \mathbf{t}, \tag{B.24}$$

and the root searching will restart again from Eq. B.14.

Although this linear solver demands more calculation during one iteration than CG solver, its robustness and efficiency on the solution of linear systems generated in ISPH algorithms, shown by the numerical experiment in §3.3, have given itself more advantages than the CG solver.

Convergence criterion

The convergence of the linear solver is achieved when the iteration number reaches the maximum iteration number, or

$$\frac{\| \mathbf{r}_i \|_2}{\| \mathbf{r}_0 \|_2} \leq \varepsilon \tag{B.25}$$

where $\| \cdot \|_2$ is the l_2-norm; subscript i and 0 are for the ith iteration and the initial value respectively; ε is the linear solver tolerance, $\varepsilon = 10^{-5}$. The discussion about the convergence of linear system can be gained from [29].

Appendix C

Publication lists

During three years PhD, the author did not only focus on the meshless SPH method, but also studied some other topics, which provides great help for the research. The publications during the PhD time are listed as follow.

Journal Publications:

R. Xu, P. Stansby, D. Laurence. Incompressible SPH method: analytical and highly accurate validation for multiphase/free-surface flows. International Journal of Numerical Method in Fluids, under review. R. Xu, P. Stansby, D. Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. Journal of Computational Physics, 228: 6703 - 6725, 2009.

E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence and P. Stansby, Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method, Journal of Computational Physics, 227: 8417-8436, 2008.

Conference Publications:

R. Xu, P. K. Stansby, D. Laurence, A stabilized Incompressible SPH method, ERCOFTAC SIG SPHERIC IV international workshop, Nantes, France, 2009.

R. Xu, C. Moulinec, P. K. Stansby, B. D. Rogers, D. Laurence, Simulations of vortex spin-down and Taylor-Green Vortices with Incompressible SPH method, ERCOFTAC SIG SPHERIC III international workshop, Lausanne, Switzerland, 2008.

P. K. Stansby, R. Xu, B. D. Rogers, et al., Modeling tsunami overtopping of a sea defense by shallow-water Boussinesq, VOF and SPH methods, Flood Risk Management Conference, Oxford, 2008 .

P.K. Stansby, A.C. Hunt, R. Xu, P.H. Taylor, A.G.L. Borthwick, T. Feng, D.R. Laurence, Wave overtopping from focused wave groups: experiments and modeling, Flood Risk Management Conference, Plymouth, 2007.

Bibliography

[1] F. Archambeau, N. Mechitoua, and M. Sakiz. A finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes, 1(1):1 - 62, 2004.

[2] T. Belytschko, Y. Krongauz, J. Dolbow and C. Gerlach. On the completeness of meshfree particle methods. International Journal for Numerical Methods in Engineering, 43: 785 - 819, 1998.

[3] D. Beeman. Some multistep methods for use in molecular dynamics calculations. Journal of Computational Physics, 20: 130 - 139, 1976.

[4] J. Bonet and T. S. L. Lok, Variational and momentum preservation aspects of smooth particle hydrodynamics formulations, Computer Methods in Applied Mechanics Engineering 43: 785 - 819, 1998.

[5] S. Børve, M. Omang and J. Trulsen. Regularized smoothed particle hydrodynamics with improved multi-resolution handling. Journal of Computational Physics, 208: 345 - 367, 2005.

[6] S. Børve and D. J. Price. Hydrodynamical instabilities in compressible fluids using SPH. Proceedings of the 4th SPHERIC Workshop, Ecole Centrale De Nantes, Nantes, France, May, 2009.

[7] D. L. Brown, R. Cortez and M. L. Minion Accurate projection methods for the Incompressible Navier-Stokes equations. Journal of Computational Physics, 168: 464 - 499, 2001.

[8] G. Y. Buss and P. K. Stansby. SAWW - a computer program to calculate the properties of steady water waves. Internal reports. Simon Engineering Laboratories, University of Manchester, 1982.

[9] A. K. Chaniotis, D. Poulikakos and P. Koumoutsakos. Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. Journal of Computational Physics, 182: 67 - 90, 2002.

[10] J. K. Chen, J. E. Beraun and T. C. Carney. A corrective smoothed particle method for boundary value problems in heat conduction. International Journal for Numerical Methods in Engineering 1999; 46(2): 231 - 252.

[11] J. K. Chen, J. E. Beraun and C. J. Jih. Completeness of corrective smoothed particle method for linear elastodynamics. Computational Mechanics 1999; 24(4): 273 - 285.

[12] A. J. Chorin. Numerical solution of the Navier-Stokes equations. Mathematics of Computation, 22: 745 - 762, 1968.

[13] A. J. Chorin. Numerical study of slightly viscous flow, Journal of Fluid Mechanics, 57: 785 - 796, 1973

[14] F. Colin, R. Egli and F. Y. Lin. Computing a null divergence velocity field using smoothed particle hydrodynamics. Journal of Computational Physics, 217: 680 - 692, 2006.

[15] A. Colagrossi and M. Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. Journal of Computational Physics, 191: 448 - 475, 2003.

[16] G.-H. Cottet and P. D. Koumoutsakos. Vortex methods: theory and practice. Cambridge Univ. Press, Cambridge, U.K., 2000.

[17] S. J. Cummins and M. Rudman. An SPH projection method. Journal of Computational Physics, 152: 584 - 607, 1999.

[18] R. A. Dalrymple and B. D. Rogers. Numerical modeling of water waves with the SPH method. Coastal Engineering, 53: 141-147, 2006.

[19] B. J. Daly. A technique for including surface tension effects in hydrodynamic calculations. J. Comput. Phys., 4: 97 - 117, 1969.

[20] M. S. Darwish. A new high-resolution scheme based on the normalized variable formulation. Numerical Heat Transfer, Part B, 24: 353 - 371, 1993.

[21] R. Das and P. W. Cleary. Modelling 3D fracture and fragmentation in a thin plate under high velocity projectile impact using SPH. ERCOFTAC SIG SPHERIC IIIrd International workshop, Lausanne, Switzland.

[22] S. F. Davis. Flux difference splittings and limiters for the resolution of contact discontinuities. Appl. Mathematics and Computation, 65: 3 - 18, 1994.

[23] R. G. Dean and R. A. Dalrymple. Water wave mechanics for engineers and scientists, Advanced Series on Ocean Engineering, vol. 2: 172 - 185, 1991.

[24] J. W. Dold and D. H. Peregrine, An efficient boundary integral method for steep unsteady water waves. In Methods for Fluid Dynamics II (ed. K. W. Morton & M. J. Baines), pp. 671 - 679. Oxford University Press, 1986.

[25] M. Ellero, M. Serrano and P. Español. Incompressible smoothed particle hydrodynamics. Journal of Computational Physics, 226: 1731 - 1752, 2007.

[26] J. Fang and R. G. Owens and L. Tachera and A. Parriaux, A numerical study of the SPH method for simulating transient viscoelastic free surface flows, Journal of Non-Newtonian Fluid Mech., 139: 68 - 84, 2006.

[27] J. Fang and A. Parriaux. A regularized Lagrangian finite point method for the simulation of incompressible viscous flows. Journal of Computational Physics, 227: 8894 - 8908, 2008.

[28] A. Ferrari, M. Dumbser, E. F. Toro and A. Armanini. A new 3D parallel SPH scheme for free surface flows. Computers & Fluids, 38: 1203 - 1217, 2009.

[29] J. H. Ferziger and M. Perić. Computational methods for fluid dynamics. Springer, 1997.

[30] R. W. Fox and A. T. Mcdonald. Introduction to fluid mechanics. John Wiley and Sons Inc., New York, 1994.

[31] M. G. Gesteria, B. D. Rogers, R. A. Dalrymple, A. J. C. Crespo and M. Narayanaswamy. User guide for the SPHysics code. SPHysics v1.4.000 guide. February, 2009.

[32] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and multigrid method, Journal of Computational Physics, 48: 387 - 411, 1982.

[33] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. Royal Astronomical Society, Monthly Notices, 181: 375 - 389, 1977.

[34] H. Gotoh, T. Shibahara and T. Sakai, Sub-particle-scale turbulence model for the MPS method - Lagrangian flow model for hydraulic engineering, Comput. Fluid Dyn. J. 9 (4): 339 - 347, 2001.

[35] H. Gotoh, S. Shao, and T. Memita. SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater. Coastal Engineering Journal, 46 (1): 39 - 63, 2004.

[36] J. P. Gray, J. J. Monaghan, and R. P. Swift. SPH elastic dynamics. Computer methods in applied mechanics and engineering, 190: 6641 - 6662, 2001.

[37] M. H. Gutknecht and M. Rozložník. A framework for generalized conjugate gradient methods—with special emphasis on contributions by Rüdiger Weiss. Applied Numerical Mathematics 41: 7 - 22, 2002.

[38] M. Hayashi, H. Gotoh, T. Memita, and T. Sakai. Gridless numerical analysis of wave breaking and overtopping at upright seawall. Proceedings of 27th International Conference on Coastal Engineering, Sydney, Australia: 2100 - 2113, 2000.

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

[40] C. Hirt and B. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics, 39(1): 97 - 117, 1981.

[41] C. Hirt, A. Amsden, J. Cook. An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds. Journal of Computational Physics, 135: 203 - 216, 1997.

[42] W. G. Hoover. Isomorphism linking smooth particles and embedded atoms. Physica A, 260: 244 - 254, 1998.

[43] W. G. Hoover, C. C. Hoover, and E. C. Merritt. Smooth particle applied mechanics: conservation of angular momentum with tensile stability and velocity averaging. Physical Review E, 69: 016702, 2004.

[44] X. Y. Hu and N. A. Adams, An incompressible multi-phase SPH method, Journal of Computational Physics, 227: 264 - 278, 2007.

[45] Y. Imaeda, T. Tsuribe and S.-I. Inutsuka. Gas accretion from the elliptic gas disk to the binary system. ERCOFTAC SIG SPHERIC IIIrd International workshop, Lausanne, Switzland, 2008.

[46] R. Issa. Numerical assessment of the smoothed particle hydrodynamics gridless method for incompressible flows and its extension to turbulent flows. PhD Thesis. University of Manchester, 2007.

[47] R. Issa, D. Violeau and G. Thibault. HPC for Spartacus-3D SPH code and applications to real environmental flows. ERCOFTAC SIG

SPHERIC IIIrd International workshop, Lausanne, Switzland: 121 - 126, 2008.

[48] H. Jasak and A. Gosman, Automatic resolution control for the finite-volume method, part 1 : A-posteriori error estimates, Numerical Heat Transfer, Part B: Fundamentals, 38: 237 - 256, 2000

[49] I. M. Janosi, D. Jan, K. G. Szabo and T. Tel. Turbulent drag reduction in dam-break flows. Experiments in Fluids, 37: 219 - 229, 2004

[50] E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - I: surface approximations and partial derivatives estimates. Computers and Mathematics with Applications, 19: 127 - 145, 1990.

[51] E.J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - II: solution to parabolic, hyperbolic and elliptic partial differential equations. Computers and Mathematics with Applications 19: 147 - 161,1990.

[52] R. Keck. The finite volume particle method: a meshless projection method for incompressible flow. Ph.D. thesis, University of Kaiserslautern, 2002.

[53] R. Keck and D. Hietel. A projection technique for incompressible flow in the meshless finite volume particle method. Advances in Computational Mathematics 23: 143 - 169, 2005.

[54] A. Khayyer. Improved Particle Methods by Refined Differential Operator Models for Free-Surface Fluid Flows. Ph.D. thesis, Kyoto University, 2008.

[55] J. Klapp, L. D. G. Sigalotti, S. Galindo and R. Durate. Numerical calculations of flow through an orifice conduit. ERCOFTAC SIG SPHERIC IInd International workshop, Madrid, Spain: 32 - 36, 2007.

[56] S. Koshizuka, H. Tamako, and Y. Oka. A particle method for incompressible viscous flow with fluid fragmentation. Journal of Computational Fluid Dynamics, 4: 29 - 46, 1995.

[57] S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of compressible fluid. Nuclear Science Engineering, 123: 421 - 434, 1996

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

[59] J. Kuhnert and S. Tiwari. Finite pointset method based on the projection method for simulations of the incompressible Navier-Stokes equations. Report in Institut Techno- and Wirtschaftsmathematik, 2001.

[60] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. Journal of Computational Physics, 113: 134 - 147, 1994.

[61] J. Larsen and H. Dancy. Open boundaries in short wave simulations - a new approach. Coastal Engineering, 7: 285 - 297, 1983.

[62] E. S. Lee. Truly incompressible approach for computing incompressible flow in SPH and comparisons with the traditional weakly compressible approach. PhD Thesis. University of Manchester, 2007.

[63] E. S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence and P. Stansby. Comparisons of Weakly Compressible and Truly Incompressible Algorithms for the SPH Mesh Free Particle Method. Journal of Computational Physics, 227: 8417 - 8436, 2008.

[64] A. Leonard. Vortex method for flow simulation. Journal of Computational Physics, 37: 289 - 335, 1980.

[65] B. P. Leonard. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. Computer Methods in Applied Mechanics and Engineering, 88: 17 - 74, 1991.

[66] E. Y. M. Lo and S. Shao. Simulation of near-shore solitary wave mechanics by an incompressible SPH method. Applied Ocean Research 24: 275 - 286, 2002.

[67] L. Lucy. Numerical approach to the testing of the fission hypothesis. The Astronomical Journal, 82: 1013 - 1024, 1977.

[68] P. A. Madsen, H. B. Bingham and H. Liu. A new Boussinesq method for fully nonlinear waves from shallow to deep water. Journal of Fluid Mechanics, 462: 1 - 30, 2002.

[69] A. J. Majda and A. L. Bertozzi. Vorticity and incompressible flow, Cambridge Texts in Applied Mathematics. Cambridge Univ. Press, Cambridge, U.K., 2002.

[70] J. J. Monaghan and J. C. Lattanzio. A refined particle method for astrophysical problems. Astronomy and Astrophysics, 149 (1): 135 - 143, 1985.

[71] J. J Monaghan. Smoothed particle hydrodynamics. Annual review of astronomy and astrophysics, 30: 543 - 574, 1992.

[72] J. J. Monaghan. Simulating free surface flows with SPH. Journal of Computational Physics, 110: 399 - 406, 1994.

[73] J. J. Monaghan and A. Kocharyon. SPH simulation of multi-phase flow. Computer Physics Communications, 87: 225 - 235, 1995.

[74] J. J. Monaghan. SPH and Riemann solvers. Journal of Computational Physics, 136: 298 - 307, 1997.

[75] J. J. Monaghan, A. Kos. Solitary waves on a cretan beach. Journal of waterway, port, coastal, and ocean engineering, 125(3): 145 - 154, 1999.

[76] J. J. Monaghan. SPH without a tensile instability. Journal of Computational Physics, 159: 290 - 311, 2000.

[77] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. Journal of Computational Physics, 136: 214 - 226, 1997.

[78] J. P. Morris. Analysis of SPH with applications, Ph.D. thesis (Monash University, Melbourne, Australia, 1996).

[79] D. Stevens, H. Power, M. Lees and H. Morvan. The use of PDE centres in the local RBF Hermitian method for 3D convective-diffusion problems. Journal of Computational Physics, 228: 4606 - 4624, 2009.

[80] R. M. Nestor, M. Basa, M. Lastiwka and N. J. Quinlan. Extension of the finite volume particle method to viscous flow. Journal of Computational Physics, 228: 1733 - 1749, 2008.

[81] R. M. Nestor and N. J. Quinlan. Incompressible moving boundary flows with the finite volume particle method. ERCOFTAC SIG SPHERIC IVth International workshop, Nantes, France: 192 - 199, 2009.

[82] G. Oger, M. Doring, B. Alessandrini and P. Ferrant, An improved SPH method: towards higher order convergence. Journal of Computational Physics, 225: 1472 - 1492, 2007.

[83] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor and C. Sacco. A stabilized finite point method for analysis of fluid mechanics problems. Computer Methods in Applied Mechanics and Engineering, 139: 315 – 346, 1996.

[84] P. A. Patsis and E. Athanassoula. SPH simulations of gas flow in barred galaxies. Effect of hydrodynamical and numerical parameters. Astronomy and Astrophysics, 358: 45 - 56, 2000.

[85] D. H. Peregrine. Long waves on a beach. Journal of Fluid Mechanics, 27: 815 – 827, 1967.

[86] D. H. Peregrine. Flow due to vertical plate movint in a channel. Internal report, private communication, 1972.

[87] S. Potapov, B. Maurel and A. Combescure. SPH framework to model fluid shell interactions. ERCOFTAC SIG SPHERIC IIIrd International workshop, Lausanne, Switzland: 162 - 167, 2008.

[88] R. K. Price, The Breaking of Water Waves, J. Geophys. Res., 76(6): 1576–1581, 1971.

[89] D. J. Price, Modelling discontinuities and Kelvin-Helmholtz instabilities in SPH, Journal of Computational Physics, 227: 10040-10057, 2008.

[90] N. Quinlan, M. Basa and M. Lastiwka. Truncation error in mesh-free particle methods. International Journal for Numerical Methods in Engineering, 66: 2064 - 2085, 2005.

[91] B. Ramaswamy and M. Kawahara. Lagrangian finite element analysis applied to viscous free surface fluid flow. International Journal for Numerical Methods in Fluids, 7: 953 - 984, 1987.

[92] C. Rehbach. Numerical calculation of three-dimensional unsteady flows with vortex sheets, AIAA paper 111 - 1978.

[93] C. M. Rhie and W. L. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA Journal, 21: 1525 - 1532, 1983

[94] M. M. Rienecker and J. D. Fenton, A Fourier approximation method for steady water waves, Journal of Fluid Mechanics, 104: 119 - 137, 1981.

[95] A. J. Roberts, Transient free-surface flow generated by a moving vertical plate, The Quarterly Journal of Mechanics and Applied Mathematics, 40: 129 - 158, 1987

[96] B. D. Rogers and R. A. Dalrymple. SPH modeling of breaking waves. Proc. 29th Intl. Conference on Coastal Engineering. World Scientific Press: 415 - 427, 2004

[97] L. Rosenhead. The formation of vortices from a surface of discontinuity. Proc. R. Soc. Lond. A 134: 170 - 191, 1931.

[98] C. Schick, Anisotropic smoothing kernels for particle methods in fluid flow, Master's thesis, University of Kaiserslautern, 2000.

[99] H. F. Schwaiger. An implicit corrected SPH formulation for thermal diffusion with linear free surface boundary conditions. International Journal for Numerical Methods in Engineering, 75: 647 - 671, 2008.

[100] S. Shao and E. Lo. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. Advances in Water Resources, 26: 787 - 800, 2003.

[101] S. Shao and H. Gotoh. Simulating coupled motion of progressive wave and floating curtain wall by SPH-LES model. Coastal Engineering Journal, 46: 171 - 202, 2004.

[102] P. K. Stansby, A. Chegini and T.C. D. Barnes. The initial stages of dam-break flow. Journal of Fluid Mechanics, 374: 407 - 424, 1998.

[103] P. K. Stansby. Private communication. 2009.

[104] D. Teleaga. A finite volume particle method for conservation laws, Ph.D. thesis, University of Kaiserslautern, 2005.

[105] D. Teleaga and J. Struckmeier. A finite-volume particle method for conservation laws on moving domains. International Journal of Numerical Method in Fluids, 58: 945 - 967, 2008.

[106] S. Tiwari and J. Kuhnert. Finite Pointset Method Based on the Projection Method for Simulations of the Incompressible Navier-Stokes Equations, Lecture Notes in Computational Science and Engineering 26: 373 - 388, 2003.

[107] S. Tiwari and J. Kuhnert. Modeling of two-phase flows with surface tension by finite pointset method (FPM), Journal of Computational and Applied Mathematics 203-2: 376 - 386, 2007.

[108] O. Ubbink. Numerical predictions of two fluid systems with sharp interfaces. PhD Thesis. Imperial College of Science, Technology and Medicine, 1997.

[109] R. Vacondio and P. Mignosa. Incompressible Finite Pointset method for free surface flow. ERCOFTAC SIG SPHERIC IVth International workshop, Nantes, France: 201 - 208, 2009.

[110] R. Vacondio, B. D. Rogers and P. K. Stansby. Smoothed Particle Hydrodynamics: approximate zero-consistent 2-D boundary conditions and still shallow water tests. Submitted to Computers and Fluids, 2009.

[111] H. A. Van Der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. S. J. Sci. Stat. Comput., 13: 631 - 644, 1992.

[112] L. Verlet. Computer experiments on classical fluids. Physical Review 159 (1) : 98 - 103, 1967.

[113] G. Viccione, V. Bovolin and E. P. Carratelli. Defining and optimizing algorithms for neighbouring particle identification in SPH fluid simulations. International Journal for Numerical Method in Fluids, 58: 625 - 638, 2008.

[114] J. P. Vila. Particle Weighted methods and Smooth Particle Hydrodynamics. M3AS, 9 (2): 161 - 209, 1999.

[115] D. Violeau and R. Issa. Modelling turbulent free surface flows with smoothed particle hydrodynamics. Ercoftac bulletin, 2004.

[116] D. Violeau and R. Issa. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. International Journal for Numerical Methods in Engineering, 53: 277 - 304

[117] D. Violeau, R. Issa, S. Benhamadouche, J. Chorda and M.-M. Maubourguet. Modelling a fish passage with SPH and Eulerian codes: the influence of turbulence closure. ERCOFTAC SIG SPHERIC IIIrd International workshop, Lausanne, Switzland: 85 - 92, 2008.

[118] W. C. Welton. Two-dimentional PDF/SPH simulations of compressible turbulence flows. Journal of Computational Physics, 139: 410 - 443, 1998.

[119] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in computational Mathematics, 4 (1): 389-396, 1995.

[120] R. Xu, P. K. Stansby and D. Laurence. Accuracy and Stability in Incompressible SPH (ISPH) Based on the Projection Method and A New Approach. Journal of Computational Physics, 228 (18): 6703-6725, 2009.

[121] SPHysics, open-source SPH free-surface flow solver, SPHysics URL: http://wiki.manchester.ac.uk/sphysics/index.php/Main_Page. Lastest modification: 28 January 2010.