

**Computational Modelling of Fluid-Solid Interaction Problems
by Coupling Smoothed Particles Hydrodynamics and the
Discrete Element Method**

Ke Wu

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
School of Civil Engineering

October, 2017

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

1. **Wu, K.**, Yang, D. and Wright, N., 2016. A coupled SPH-DEM model for fluid-structure interaction problems with free-surface flow and structural failure. *Computers & Structures*, 177, pp.141-161.

Author Contributions: D.Y. conceived and designed the experiments; K.W. performed the experiments; D.Y., K.W. and N.W. analyzed the data; D.Y., K.W. and N.W. wrote the paper.

2. Yang, D., **Wu, K.**, Wan, L. and Sheng, Y., 2017. A Particle Element Approach for Modelling the 3D Printing Process of Fibre Reinforced Polymer Composites. *Journal of Manufacturing and Materials Processing*, 1(1), p.10.

Author Contributions: D.Y. conceived and designed the experiments; K.W. performed the experiments; D.Y., K.W., L.W. and Y.S. analyzed the data; Y.S. contributed analysis tools; D.Y., K.W., L.W. and Y.S. wrote the paper.

3. **Wu, K.**, Yang, D., Wright, N. and Khan, A., 2018. An integrated particle model for fluid– particle–structure interaction problems with free-surface flow and structural failure. *Journal of Fluids and Structures*, 76, pp.166-184.

Author Contributions: D.Y. conceived and designed the experiments; K.W. performed the experiments; D.Y., K.W., N.W. and A.K. analyzed the data; D.Y., K.W., N.W. and A.K. wrote the paper.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

The right of Ke Wu to be identified as Author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

© 2017 The University of Leeds and Ke Wu

Acknowledgements

I am grateful to start my PhD journey at School of Civil Engineering, the University of Leeds, UK. Indeed, from the past four years, I have received tremendous support, help and guidance from the school to focus on my research. At this point, I would like to sincerely express my thanks to everyone in the university, who helped me a lot during the course of completing this thesis.

I would like to thank my principle supervisor Dr. Dongmin Yang for accepting me into his research group. Being under your supervision for these four years has brought me into a level that I can individually discover, study and analyse various problems in different angles. The detailed comments and suggestions on journal papers have improved my scientific writing skill a lot. I would appreciate that you financially supported me to attend international conferences and my work presented orally has drawn massive attention from other research community. I will also not forget your wife, Shanshan, who cooked many yummy Chinese food to make me feel at home in UK. Thank you very much.

I will thank my co-supervisors Prof. Nigel Wright and Dr. Amirul Khan, who gave me valuable advices and suggestion during the period of supervision. You broadened my horizon by teaching my how to interpret scientific data and showing me some cutting-edge ideas.

Finally, I'd like to thank my family. This achievement would not have been possible without your support and understanding. I regret not celebrating

Chinese New Year with you once since I studied abroad in 2008. Dear dad and mom, I love you forever.

To everyone who turned up in my life, whose name unfortunately I can't display all here, you mean more to me than you will ever know. I appreciate you all.

Abstract

Discrete Element Method (DEM) and Smoothed Particles Hydrodynamics (SPH) are integrated to investigate the macroscopic dynamics of fluid-solid interaction (FSI) problems. This coupled model is originated from two different meshless methods without mesh generation, which can handle fluid-particle-structure interactions with structural deformation/failure. With SPH the fluid phase is represented by a set of SPH particle elements moving in accordance with the Navier-Stokes equations. The solid phase consists of single or multiple solid particle(s) phase and deformable structure(s) phase which are represented by DEM particle elements using a linear contact model and a linear parallel contact model to account for the interaction between particle elements, respectively. To couple the fluid phase and solid particle phase, a local volume fraction and a weighted average algorithm are proposed to reformulate the governing equations and the interaction forces. The structure phase is coupled with the fluid phase by incorporating the structure's DEM particle elements in SPH algorithm. The interaction forces between the solid particles and the structure phases are computed using the linear contact model in DEM. The proposed model is capable of simulating simultaneously fluid-structure interaction, particle-particle interaction and fluid-particle interaction, with good agreement between complicated hybrid numerical methods and experimental results being achieved. Finally, two engineering problems in injection moulding and 3D printing process are carried out to demonstrate the capability of the

integrated particle model for simulating fluid-solid interaction problems with the occurrence of structural failure.

Table of Contents

Acknowledgements	iv
Abstract	vi
Table of Contents	viii
List of Tables	xi
List of Figures	xii
1 Introduction	1
1.1 Background	1
1.1.1 Numerical simulation of Fluid-Solid Interaction	1
1.1.2 Mesh-based methods.....	4
1.1.3 Meshfree methods	8
1.2 Smoothed Particle Hydrodynamics	9
1.2.1 Development of SPH.....	9
1.2.2 Recent applications of SPH	11
1.2.3 Advantages and limitations of SPH.....	12
1.3 Discrete Element Method	14
1.3.1 Development of DEM	14
1.3.2 Recent applications of DEM.....	16
1.3.3 Advantages and limitations of DEM	17
1.3.4 PFC2D 5.0	19
1.4 Numerical coupled modelling	21
1.5 Motivations and objectives	27
1.6 Thesis outline	29
2 SPH for fluid dynamics	30
2.1 Navier-Stokes equations in SPH form.....	30
2.1.1 Interpolation of a function and interpolation of the derivative of a function	31
2.2 Numerical implementation of SPH	34
2.2.1 Kernel selections	34
2.2.2 Physical Viscosity.....	35
2.2.3 Tensile instability.....	36
2.2.4 Boundary Treatment	39
2.2.7 Time Integration	41

3 DEM for granular flow and structure mechanism	42
3.1 Principles of DEM.....	42
3.1.1 Force-displacement law	42
3.1.2 Law of motion	45
3.2 Contact models	46
3.2.1 Linear contact model.....	46
3.2.2 Linear parallel bond model.....	47
3.3 Fracture criteria of structure	49
3.3.1 Particle distribution	49
3.3.2 Bond stiffness determination	50
3.3.3 Bond ultimate strength determination	51
4 Coupling schemes for fluid-solid interaction	54
4.1 Interaction forces for fluid particle in SPH	57
4.1.1 Local averaging technique	57
4.1 Interaction forces for solid particle in DEM.....	59
4.2 Interaction forces for fluid in SPH.....	63
4.3 Interaction forces for structure in DEM.....	65
4.4 Flowchart of computational algorithms in SPH-DEM	66
5 Validations of SPH.....	68
5.1 Static tank test.....	68
5.2 Dam break test	72
5.2.1 Introduction	72
5.2.2 Effect of different smoothing lengths.....	75
5.2.4 Effect of different particle resolutions	79
6 Validations of DEM.....	82
6.1 Dry dam break test	82
6.2 Tip-loaded cantilever beam test	84
7 Validations of SPH-DEM	90
7.1 Single particle sedimentation test.....	90
7.2 Two phase dam-break test.....	94
7.3 Dam break with top-fixed elastic gate	100
7.4 Dam break with bottom-fixed elastic gate	106
7.5 Two phase dam-break test with bottom-fixed elastic gate	111

8 3D Printing of Fibre Reinforced Polymer Composites.....	115
9 2D simulation of injection moulding process of short fibre composites	127
10 Conclusions and future works	144
10.1 Conclusions.....	144
10.2 Future works.....	147
List of References	148
Appendix 1: Publication list	169
Appendix 2: The main structure of SPH code in C++	170

List of Tables

Table 1 SPH parameters used for the static tank test.....	70
Table 2 SPH parameters for the dam-break test.....	75
Table 3 Particle resolutions in the dam-break test	79
Table 4 The list of material and particle properties	84
Table 5 Deflections for the tip-loaded cantilever beam test	86
Table 6 Maximum applied load for the tip-loaded cantilever beam test	89
Table 7 Parameters for SPH-DEM modelling of the elastic gate test.....	103
Table 8 Numerical parameters in SPH-DEM modelling of injection moulding process	132

List of Figures

Fig. 1 Particle approximations for particle i within the support domain kh of the kernel function W . r_{ij} is the distance between particle i and j , s is the surface of integration domain, Ω is the circular integration domain, k is the constant related to kernel function and h is the smooth length of kernel function.	32
Fig. 2 Truncation of particle support domain by a boundary.....	40
Fig. 3 Boundary particles and their interaction with SPH particles	40
Fig. 4 Two particles in direct contact	43
Fig. 5 2D representation of a contact between two particle elements in DEM	47
Fig. 6 DEM particle elements with a parallel bond	48
Fig. 7 Hexagonal packing of discrete particles with parallel bonds.....	50
Fig. 8 Schematic diagram of interaction forces in the integrated particle model.....	56
Fig. 9 Computational flow chart of the integrated particle model	67
Fig. 10 Initial configuration of the static tank test.....	69
Fig. 11 Particle distribution during a period time of 1.0s using two different kernel functions.	72
Fig. 12 2D SPH representation of the dam-break test.....	73
Fig. 13 Initial density of SPH particles with an assumption of artificial compressibility	74
Fig. 14 Results from experiment [51], MPS [53] and SPH with $h=1.25 \times \Delta p$ for a time period of $t=1.0s$	77
Fig. 15 SPH simulations with three different smoothing length for a time period of $t=1.0s$	78
Fig. 16 SPH simulations with three different particle resolutions for a time period of $t=1.0s$	80
Fig. 17 Dry dam break test for a time period 0.5 s.	83
Fig. 18 Configuration of cantilever under single point load in DEM.....	85
Fig. 19 Distribution of stress σ_{11} in cantilever beam at load 5000N.....	87
Fig. 20 Configuration of single particle sedimentation test.....	91
Fig. 21 Longitudinal coordinate (a) and velocity (b) against time	93
Fig. 22 2D representation of the two phase dam-break test	96
Fig. 23 Two phase dam-break test for a time period of $t=0.2s$	98

Fig. 24	The normalised front position against the characteristic time	100
Fig. 25	Configuration of 2D elastic gate test in a coupled SPH-DEM model.....	101
Fig. 26	Comparisons between experimental, SPH-SPH and SPH-DEM results of elastic gate test with a time period of 0.40s.	104
Fig. 27	Water levels at different time	105
Fig. 28	2D representation of FSI with fracture	108
Fig. 29	SPH-DEM modelling of FSI with fracture	111
Fig. 30	Configuration of the dam-break test with Fluid-Solid particle-Structure interaction.....	113
Fig. 31	SPH-DEM modelling of FSI with fracture	114
Fig. 32	Initial configurations of the 3D printing processes of composites. (a) Short glass fibre reinforced acrylonitrile–butadiene–styrene (ABS) composite; (b) Continuous carbon fibre reinforced Nylon composite.....	120
Fig. 33	Velocities of resin at printing time of $t = 0.05$ s and $t = 0.1$s....	123
Fig. 34	Density distribution in resin at printing time of $t = 0.05$ s and $t = 0.1$ s.....	125
Fig. 35	Tensile stress in continuous carbon fibre/Nylon: (a) $t = 0.05$ s and (b) $t = 0.1$ s.	126
Fig. 36	Configuration of 2D SPH-DEM simulation of the injection moulding process	131
Fig. 37	Snapshots of the injection moulding process at different time	133
Fig. 38	Velocity contour of resin flow before split flow at a 0.005s time interval	134
Fig. 39	Velocity contour of resin flow after split flow at 0.01s time interval	135
Fig. 40	Front flow observed at time 0.025s	137
Fig. 41	Distribution of tensile stresses in short fibers at time 0.2s	138
Fig. 42	Injection moulding processes with fibre volume fractions of 3.8% and 8.3%.	141
Fig. 43	Qualitative comparison between (a) numerical prediction and (b) post-processed X-ray CT image of fibre orientation upon the completion of filling process	141

1 Introduction

1.1 Background

1.1.1 Numerical simulation of Fluid-Solid Interaction

Fluid-solid interaction is a branch that combines fluid mechanics and solid mechanics together to investigate the dynamic behaviour of solid under the action of fluid field or the effect of dynamic behaviour of solid to the fluid field. The most important characteristic of fluid-solid interaction is the interaction between two different phases, non-rigid solid (e.g. structure phase) can deform or rigid solid (e.g. single or multiple solid particle phase) move under the forces from fluid field and in return its deformation for structure phase and movement for structure and solid particle phases influence fluid field. Consequently fluid-solid interaction under different conditions can produce various phenomenon [1].

As fluid-solid interaction intersects various disciplines like fluid mechanics, solid mechanics, dynamics and computational mechanics, it technically has a close relation to the field of engineering such as civil, aerospace, marine, ocean, mechanical, nuclear, geology and biology [1]. For example, FSI applications include, but are not limited to, sedimentation [2-4], particle assembly [5], aerodynamics [6, 7], turbulence [8, 9], complex flows in irregular domains [10-12], electro-hydrodynamics [13], magneto-hydrodynamic flows [14], biofluid and bio-mechanics [15, 16]

Due to the complication of fluid-solid interaction, the research and development of fluid-solid interaction advances slowly in recent decades and

it is only limited to analytical solutions and laboratory experiments. For most FSI problems, analytical solutions are impossible to solve model equations and laboratory experiments are also limited in scope, dependent on the scale of experiment and the massive financial investment; thus to investigate the fundamental physics involved in the complex interaction between fluid and solid, numerical simulations as an alternative solution may be employed [17]. Fluid-solid interaction gradually draws worldwide attention until the presence of numerical methods (e.g. boundary element and finite element methods) in the 1960 [18]. Later on, the fast development in computer technology and numerical methods boosts the research in fluid-solid interaction.

FSI problems usually involve flow nonlinearity and multiphysics which are too complex to be solved by analytical methods, and a small number of numerical models have been developed in recent years. Although there are various numerical methods being developed and applied to simulate the separate behaviour of fluid and solid, combined methods for FSI are still limited. The challenge of coupling two methods for FSI largely depends on the nature of their discretisation. Conventional mesh-based methods such as the finite difference method (FDM), the finite element method (FEM) and the finite volume method (FVM) discretise the domain into individual meshes. The reliance on mesh makes the treatment of discontinuities (e.g., wave breaking, cracking and contact/separation) difficult because the path of discontinuities may not coincide with the mesh lines. Remeshing techniques can ensure the discontinuities evolve along the mesh lines but at the expense of reduced computational efficiency and degradation of numerical accuracy. In comparison to conventional mesh-based methods, meshfree (or

meshless) methods are intended to approximate mathematic equations in the domain only by nodes without being connected by meshes. If the nodes are particles that carry physical properties (e.g., mass) and the system is simulated by the evolution of the particles' trajectory and the particles' properties, then this type of method is usually called a particle method. Typical particle methods are molecular dynamics (MD), Discrete Element Method (DEM), Smoothed Particle Hydrodynamics (SPH), Immersed Particle Method and Lattice Boltzmann Method (LBM). It should be noted that in LBM the particles are only allowed to move along the predefined lattices, so it is in some ways a mesh-based particle method. In the meshfree particle methods of MD, SPH and DEM a contact detection algorithm as well as an interaction law is required to define the particle interaction. The contact detection algorithm is used to determine whether two particles are interactive, and once they interact, then the interaction law must be used to calculate the interaction forces. In previous research, LBM and SPH are mainly used for simulating fluid flow [19, 20] whilst DEM is mainly used for simulating granular flow [21] and solid fracture [22]. Coupled models like SPH-SPH [23], SPH-DEM [24], Immersed Particle Method [25, 26] and LBM-DEM [27] have been developed for fluid-structure or fluid-particle interactions.

1.1.2 Mesh-based methods

The description of the physical governing equations can be divided into two fundamental frames: the *Eulerian* description and the *Lagrangian* description. The *Eulerian* description based on spatial distribution is a way of looking at fluid motion that focuses on specific locations in the space through which the fluid flows as time passes, and is typically represented by the Finite Difference Method (FDM) and Finite Volume Method (FVM) [16, 28]. The *Lagrangian* description is in a material description that looking at fluid motion where the observer follows an individual fluid parcel as it moves through space and time, and is typically represented by the finite element method (FEM) [29, 30]. Accordingly, the *Eulerian* and *Lagrangian* descriptions correspond to two different grids of domain discretization: the *Eulerian* grid and the *Lagrangian* grid.

In the *Lagrangian* grid-based methods (e.g. FEM), the grid is linked to be part of the material in the entire computation process, and therefore it moves as long as the material moves. Since the grid point in the material connects the intersected grid nodes, the relative movement of the connecting nodes may be expanded, compressed or deformed to change the shape of mesh cell. Consequently mass, momentum and energy are transported with the movement of the mesh cells.

The Lagrangian grid-based methods have several advantages.

1. No matter how the material moves, the grid is not detached on the moving material, all the field variables at a material point can be easily tracked and achieved in the entire physical time domain.

2. As some grid nodes can be placed along boundaries and material interfaces with fixed position or pre-defined velocity, the boundary conditions at free surfaces, no-slip/moving boundaries, and material interfaces can be easily implemented.
3. Depending on the geometry of material, regular or irregular mesh can be conveniently applied.
4. Since the problem domain is the only interest of area that requires the creation of grid, no additional grids beyond the problem domain are needed, as a result, the *Lagrangian* grid-based methods are efficient in computational cost.

Due to these advantages listed above, *Lagrangian* methods are widely applied to solve computational solid mechanics (CSM) problems, where the deformation is not significant as that in the fluid flows. However, when considering large distortion of mesh, *Lagrangian* grid-based methods are practically very difficult to guarantee an acceptable accuracy, the accuracy of solution is heavily affected by the formulation stored in the mesh. A feasible remedy to optimize the *Lagrangian* computation is to rezone the mesh or re-mesh the problem domain. The mesh rezoning normally can be an overlaying of a new and undistorted mesh to replace the old, distorted one, so that the computation in the next step can be carried out with the application of new and undistorted mesh. As a result, mesh rezoning is a common solution for numerical modelling of metal forming processes [31], singular problems [32] and FSI problems [33]. However, the procedure for mesh rezoning in *Lagrangian* computations is much less efficient as it spends the majority of time in mesh rezoning.

In comparison to the *Lagrangian* grid, the *Eulerian* grid doesn't spatially move and physically deform with time in the problem domain. In the same time, the material (e.g. fluid) flows across the fixed mesh cells from one end to another end. As the *Eulerian* grid and the material are treated separately, no matter how large the material deforms, there is no influence to the *Eulerian* grid, hence the numerical problems appeared in the *Lagrangian* grid is not existed in the *Eulerian* grid. Due to this apparent benefit, *Eulerian* methods are increasingly recognised and widely applied in the field of Computational Fluid Dynamics (CFD). Some early researchers has used *Eulerian* methods to investigate explosion and high velocity impacts problems with large deformation of materials [34, 35]. Contrary to the advantages in *Eulerian* methods, the following are disadvantages in *Eulerian* methods.

1. Due to the nature of *Eulerian* methods, the movement of material can not be tracked in the *Eulerian* grid. It's only able to capture the time history of field variables instead.
2. Mesh regeneration is normally required for irregular and complicated geometries. It is expensive in the entire process of numerical simulation.
3. It's difficult to accurately deal with free surface, moving boundary and the interface between the moving materials as the grid point in *Eulerian* methods is fixed.

In order to take the advantages as well as avoid the disadvantages in both *Lagrangian* and *Eulerian* descriptions, the development of coupling these two methods has been made by proposing the Coupled Eulerian Lagrangian

(CEL) [34] and the Arbitrary Lagrange Eulerian (ALE) [36]. The CEL approach is to define the different regions of the computational domain in both the *Eulerian* and *Lagrangian* methods separately. For example, the solid phase is represented by *Lagrangian* methods and *Eulerian* method accounts for the fluid phase and then these two regions continuously interact with each other by exchanging computational information through an appropriate coupling technique. When using ALE method, the mesh inside the domains can move arbitrarily to optimize the shapes of elements, while the mesh on the boundaries and interfaces of the domains can move along with materials to precisely track the boundaries and interfaces of a multi-regional system. The combination of these two approaches has drawn massive research interest and advanced with reliable solutions for FSI problems [37-39]. However, some researchers still pointed out that both the CEL and ALE are still not stable in the numerical simulations [35, 36]. An alternative approach is highly desired.

1.1.3 Meshfree methods

Meshless methods were initially proposed to overcome the difficulties associated with mesh-based methods. The main difference between meshfree and mesh-based methods is the way of approximation, in which meshless methods are dependent on nodes only. The first and typical meshless method is the Smooth Particle Hydrodynamics (SPH) [40, 41] which is initially proposed to investigate problems in astrophysics and, later on, in fluid dynamics [20, 42, 43] and even in solid mechanics [44]. In addition, many improvements have been made to minimize tensile instability [45, 46] and inconsistency [47]. The moving particle semi-implicit (MPS) method [48, 49] is similar to SPH, but the MPS method applies simplified differential operator models solely based on a local weighted averaging process without taking the gradient of a kernel function. Even though it is also applicable to fluid dynamics [50], non-conservation of momentum and spurious pressure fluctuation are the main shortcomings in MPS [51]. In the field of solid mechanics, the Discrete Element Method (DEM) [52] was initially designed to compute the motion and effect of a large number of small particles flow, later on, it was extended to study structure phase by connecting discrete particle through a bond to represent the inherent part of structure phase [22, 53]. Hence there are some major advantages of meshless methods that are more flexible than meth-based methods:

1. It's straightforward to deal with surface problems with moving discontinuities.
2. Large deformation in fluid and solid phases can be easily handled.
3. No mesh generation and sensitivity required.

1.2 Smoothed Particle Hydrodynamics

1.2.1 Development of SPH

Smoothed Particle Hydrodynamics (SPH) is a completely meshfree particle method originally invented to solve astrophysical problems in universe [40, 54], Gradually it moves to the study of fluid flow (e.g. liquid and gas) since it can be simulated by approximating the governing equations of the classical Newtonian hydrodynamics.

In the early stage of SPH, its algorithms were based on the probability theory. Even though the conservation of linear and angular momentum was not obtained under those algorithms, such acceptably promising results can be given for many astrophysical phenomena. When dealing with fluid dynamics and solid mechanics problems, the partial differential equations contributed to govern fluid dynamics and solid dynamics are difficult to devotedly reproduce [42]. Later on, SPH was rapidly developed by digging out more attractive features and identifying some inherent drawbacks. Many researchers have been devoted to improving the accuracy of SPH by proposing different remedies. For example, Gingold and Monaghan showed that a simple form of estimation leads to a particle method which does not require a grid and satisfies the conservation laws very accurately [55]. Hu and Adams showed that a simple angular-momentum conservative formulation of the viscous force can be derived theoretically under the condition of incompressible flow to conserve angular momentum [56]. What is more, Swegle et al. carried out a von Neumann stability analysis of the SPH algorithm to identify the criterion for stability or instability in terms of the stress state and the second derivative of the kernel [57]. Morris found out

that the inconsistency is the main reason to give rise to inappropriate accuracy of SPH [46]. With the identification of inherent drawbacks, some remedies have been proposed to remain the consistency and to optimize the accuracy of the SPH method. Monaghan used a symmetrized form in pressure gradient to construct symmetric forms of the SPH momentum equations, which was tested to have better accuracy [41]. Johnson considers the effects of artificial viscosity and verified that it can have a significant effect on the results [58]. Randles and Libersky extended the conservative smoothing into multi-dimensional SPH and implemented generalized boundary conditions in SPH to improve results considerably [59]. Chen et al. proposed a corrective smoothed particle method (CSPM) by solving Burgers' equations over a wide range of the viscosity parameters in one and two space dimensions to improve the simulation accuracy [60]. Batra et al. developed modified SPH (MSPH) to analyze shear strain localization in elasto-thermo-viscoplastic materials that exhibit strain- and strain-rate hardening and thermal softening [61]. The stress-point approach was developed to address tension instability and improve accuracy in Smoothed Particle Hydrodynamics (SPH) methods [62]. It has been shown that the consistency of the SPH method can be improved to acceptable levels by substituting moving least-squares (MLS) and MLS severely retards tension instability growth [63]. The kernel function can be corrected to enforce the consistency conditions and improve the accuracy [64], Chen et al. proposed a reproducing kernel particle method (RKPM) by introducing a material kernel function and an RKPM material shape function for large deformation analysis of non-linear structures [65].

1.2.2 Recent applications of SPH

SPH as a meshfree method has advantages in dealing with moving boundary problems, especially extensive deformation and free surface flow. Despite of the initial applications of the SPH method in astrophysical phenomena, the applications of SPH now have been extensively expanded to many engineering applications include:

1. In the multi-phase flows [66, 67] where fluid and gas coexisted, SPH has successfully applied into incompressible/compressible flow by solving equation of state. Incompressible flow can also be solved by using Bernoulli equation to achieve pressure and velocity fields.
2. In the aspect of coastal engineering [68, 69], SPH can be used to model the slip of solid block into water. Many interesting phenomenon can be captured like the splash of water, the appearance of vortex, the spread of wave and the pressure distribution of water.
3. For blood flow [70], it is computationally costly to compute the viscosity term in Navier-Stokes equations. In SPH, the entire blood system can be a discrete space with multi particles, then the fluid speed is acquired through solving blood flow control equations with deployment of the SPH model. On the basis of SPH simulation of fluid, the shape constraining method can help control the movement blood particles so as to obtain the solid speed. It finally avoids the complex computation process of the traditional methods and achieves fast blood flow simulations results and meets the real-time experiments.
4. For Heat and mass flow [71, 72], the second order derivative in SPH is constructed though Taylor series expansion and 1D, 2D and 3D

heat conduction problems are computed and compared with analytical solutions.

5. In the research of Ice dynamics [73, 74], it is mainly focused on floating ice and two-phase flow. When applying finite difference method and finite volume method to sea-ice interaction, the moving ice can result in the deformation of generated mesh and consequently numerical diffusion is significant to influence numerical accuracy. Under the application of particle methods, these difficulties can be easily avoided. It can model not only the movement of floating ice, but also the phase transfer between sea and ice by considering thermodynamics.
6. In the process of metal forming [64, 75], the simulation of free surface based on SPH can improve numerical accuracy.

1.2.3 Advantages and limitations of SPH

For fluid dynamics problems, SPH is a very powerful method for CFD problems governed by the Navier-Stokes equations. To model such complex free surface flows is a tough and challenging task for most computational fluid dynamics (CFD) solvers which work in the Eulerian framework. As a Lagrangian and meshless method, SPH offers a convenient tracking for different complex boundaries and a straightforward satisfaction for different boundary conditions. Therefore SPH is robust in modelling complex hydrodynamic problems characterized by free surface boundaries, multiphase interfaces or material discontinuities. Along with the rapid development of the SPH theory, related numerical techniques and high-

performance computing technologies, SPH has not only attracted much attention in the academic community, but also gradually gained wide applications in industrial circles. All in all, SPH has some special advantages over the traditional grid based numerical methods when dealing with fluid dynamics:

1. SPH is a particle method based on Lagrangian scheme, therefore the time history of the material particles can be achieved.
2. SPH can be easily implemented for modelling free surface and interfacial flow problems.
3. SPH does not require mesh generation. Hence a straightforward handling of large deformations is allowed, unlike mesh remedies applied to mesh-based methods when considering large deformations.
4. SPH can more naturally and physically develop a problem domain than mesh-based methods, especially in free surface flow.

For the limitations in SPH, small time steps required due to the use of the weakly compressible Tait equation of state, so large scale simulations using SPH have so far been rare and only performed on very expensive CPU-based supercomputers.

1.3 Discrete Element Method

1.3.1 Development of DEM

In the recent decades, the Discrete Element Method (DEM) has played a vital role in numerically modelling the dynamic behaviour of granular flow. Principally the granular flow is made up of a cluster of discrete particles, each particle is governed by two fundamental laws: law of motion and force-displacement law. Law of motion is applied to each particle and force-displacement law is applied to each contact between particle pairs. In comparison with mesh-based methods (e.g. FEM), DEM is more suitable to model large deformation and crack propagation as discrete particles in DEM are not limited to deformation and displacement.

DEM was first proposed by Cundall [76] who adopted this method to investigate large scale movement in blocky rock systems. With the development of DEM, it has been one of the dominant numerical tools in the solid mechanics and geotechnical engineering. In the process of evolving DEM, circular or spherical particles in 2D and 3D are initially assumed for easy implementation and computational simplicity. As the obtained contact forces among those regular shaped particles are only transferred through the particle centroid, there is no contribution to moment, which is totally different to irregular shaped particles. Therefore, the real dynamic behaviour of material can not be physically and accurately obtained when the real particle shape is not regular. even when the particle shape is having little differentiation from being circular or spherical [77]. Up to now, many researchers has developed many solutions to account for the irregularity of particle shape. Boon presented a two-stage contact detection algorithm for

polygonal (2-D) or polyhedral (3-D) convex particles, firstly the contact point is identified and then the contact normal is calculated from the gradient vector of an inner potential particle whose corners are rounded [78]. John F. Peters defined the surface of poly-ellipsoid particle by joining octants of eight ellipsoids, so that the particle mass, centroid and moment of inertia tensor can all be sorted out in closed form in despite of the complexity of the particle shapes [79]. More researches relevant to elliptic problems can be referred to [80, 81].

For large-scale simulation of discrete particle flow, the application of contact searching algorithm is highly sensitive to the computational efficiency, which towards to the cost of computation. Feng introduced the augmented spatial digital tree (ASDT) algorithm which is based on the understanding existing spatial digital tree-based contact detection approaches and the alternating digital tree (ADT) algorithm, it is proved to be at least over 3.9 times faster than the ADT [82]. Mio found out an optimum cell condition, which is related between cell size and particle radius, therefore it is possible to model large-scale DEM simulation [83]. For similar particle size ratio, a contact detection algorithm with no binary search was proved not to be influenced in terms of performance and additionally it technically requires less computing memories [84]. Williams presented a new spatial reasoning algorithm that can be used in multi-body contact detection. The partitioning of N bodies of arbitrary shape and size into N lists in order $O(N)$ operations were achieved under this algorithm scheme, where each list consists of bodies spatially near to the target object. What is more, the algorithm has been tested for objects of arbitrary shape and size, in two and three dimensions [85].

The determination of timestep in the DEM simulations is a key factor for the numerical accuracy and stability. If a timestep size that is too small, it results in an unnecessarily long simulation time. On the contrary, a large timestep size that is too large can give rise to incorrect simulation results in terms of numerical accuracy and stability. Initially Cundall [52] applied an computationally efficient, explicit, central difference time integration scheme, but this scheme is limited to a small enough timestep in order to be conditionally stable. In purpose of avoiding the stability issue, many researchers applied implicit time integration scheme [72, 86, 87], but these schemes are computationally costly and a large amount of iterations is necessary. In addition, Farhat combined explicit and implicit schemes together in the FSI problems to proceed them separately [88]. O'Sullivan introduced a new, simple approach for calculating the critical time increment in explicit discrete element simulations. Using this approach can be conservative and eliminate the overestimation of the actual critical time step [89].

1.3.2 Recent applications of DEM

DEM can not only apply to particulate flow, but also the deformation/fracture of material. The applications of DEM now have been extensively expanded to many engineering applications include but not limited to:

1. In the pharmaceutical industries [90, 91], The pharmaceutical powder and tableting process is simulated using a combined finite discrete element method and contact dynamics for irregular-shaped particles. The particle-scale formulation and two-stage contact detection

algorithm which has been developed for the proposed method enhances the overall calculation efficiency for particle interaction characteristics.

2. In the field of Mining [92, 93], the deformation and failure mechanism of footwall slope was simulated with the Discrete Element Method taking the effect of the water in fissures into account.
3. For powder metallurgy [94, 95], the Discrete Element Method was carried out to analyse the densification process of iron powders with conventional cold pressing, warm compaction and die wall lubrication. The powder metallurgy compaction process can be simulated exactly with the the Discrete Element Method.
4. The discrete element method has been used to model railway ballast [96, 97], Particles have been modelled using both spheres and clumps of spheres. A simple procedure has been developed to generate clumps which resemble real ballast particles much more so than spheres. The influence of clump shape on the heterogeneous stresses within an aggregate has been investigated, and it has been found that more angular clumps lead to a greater degree of homogeneity.

1.3.3 Advantages and limitations of DEM

Due to the nature of DEM, it is capable of analysing multiple interacting continuous (e.g. material), discontinuous (e.g. granular flow) or deformable interacting bodies undergoing large displacements and rotations. As a robust

and promising numerical tool, the advantages of DEM are summarised as follow:

1. Straightforward implementation in discontinuous material modelling,
2. The contact force between particles can be in any type (e.g. direct contact force or long-range interaction force),
3. Particle is not limited to a regular shape, any shape of particle can be considered to mimic any real engineering problem,
4. DEM can be easily coupled with mesh-based methods (e.g. CFD-DEM) and meshfree methods (e.g. SPH-DEM) for the existence of both continuous and discontinuous properties.

For the limitations of DEM,

1. The maximum number of particles, and duration of a virtual simulation is limited by computational power. Typical flows contain billions of particles, but contemporary DEM simulations on large cluster computing resources have only recently been able to approach this scale for sufficiently long time (simulated time, not actual program execution time).
2. DEM is computationally demanding, which is the reason why it has not been so readily and widely adopted as continuum approaches in computational engineering sciences and industry. However, the actual program execution times can be reduced significantly when graphical processing units (GPUs) are utilized to conduct DEM simulations, due to the large number of computing cores on typical GPUs. In addition GPUs tend to be

significantly more energy efficient than conventional computing clusters when conducting DEM simulations i.e. a DEM simulation solved on GPUs requires less energy than when it is solved on a conventional computing cluster.

1.3.4 PFC2D 5.0

As represented by Cundall and Strack [98], the commercial software PFC2D 5.0 code made use of the features of the distinct element method (DEM) to simulate the movement and interaction among circular particles in 2D simulation. There are few assumptions embedded in the PFC2D 5.0 computational algorithms:

1. The particles possess a rigid body feature, even when two particles have a minor overlap,
2. Under the direct contact condition, the contact is active only when a small overlap between two particles is detected. For long-range interaction, the contact model is activated as long as the distance between two particle is smaller than the pre-defined threshold value.
3. The contact force is determined by the magnitude of the overlap between two particles and the force-displacement law.
4. Bonds are introduced to connect particles as an assembly to represent the inherent material property.
5. Complex shape of particle can be created by clumping multiple circular particles as a rigid body.

In addition to dynamic behaviour of circular particles, PFC2D 5.0 has been extended to cover thermal dynamics that the simulation of transient heat

conduction and storage in materials is allowed. In a similar way, thermal contact is associated with mechanical contact when mechanical contact is active by default. Furthermore, Computational Fluid Dynamics (CFD) has been successfully embedded in PFC2D 5.0 to carry out multi-physical problems. The most exciting feature in PFC2D 5.0 is the introduction of C++ Plug-ins to enable users flexibly create any FISH intrinsics and contact models in C++. Plug-ins are compiled as Dynamic Link Library (DLL) files by using Microsoft Visual Studio 2010 SP1. In addition to flexibility, C++ Plug-ins have several advantages over FISH code:

1. C++ functions can be generally directly executed without transferring C++ functions to FISH functions. Therefore, the computational efficiency is massively improved and it is proved to be 10 to 100 times faster than FISH functions.
2. The computational efficiency can be further improved by using concurrent programming, in which several computations are executed during overlapping time periods.
3. Direct access is provided to internal data structures and methods that are not available via predefined FISH intrinsics or contact models.
4. C++ plug-in is accessible to any C++ library or DLL, but not only limited to FISH libraries.

Due to this feature, the author wrote an unique SPH code in C++ and then it is embedded into PFC2D 5.0 to couple with DEM as an integrated SPH-DEM model. The main structure of SPH code in C++ is displayed in the section of Appendix 2.

In PFC2D 5.0, the contact force is computed via force-displacement law first and then movement of particles and walls is updated accordingly through the law of motion in every calculation cycle. In the next calculation cycle, the contacts between particle and particle or particle and wall are updated in accordance with the current position of particle and wall and then the rest of the work in the new cycle is repeated by making use of the force-displacement law and the law of motion.

1.4 Numerical coupled modelling

Numerical coupled modelling has been a common method to deal with multi-physical problems. In this way, the different subsystems which form a coupled problem are modelled and simulated in a distributed manner. Furthermore, the coupled simulation is carried out by running the subsystems separately and the subsystems will exchange data accordingly during the entire process of the simulation.

As FSI involves two phases, *i.e.*, fluid and solid particle/structure, the numerical methods for each can be the same or different. As the interface between the fluid and solid phases is evolving in space and time, the numerical models of FSI can be classified as *Eulerian-Eulerian*, *Eulerian-Lagrangian* and *Lagrangian-Lagrangian*. In general, an *Eulerian* method discretises the space into a mesh and defines the unknown values at the fix points, while a *Lagrangian* method tracks the pathway of each moving mass point. Communications between the mathematical frameworks for fluid and structure are realised through a fluid-structure interface.

The *Eulerian-Eulerian* models tend to use an *Eulerian* FDM to treat both fluid and structure boundaries on fixed meshes to avoid mesh reconstruction. This is able to handle large deformation and free movement of the structure in the fluid as well as the contact between structures. However, this comes at the price of high computational costs and additional discretisation errors since the interface is only tracked implicitly by the solution itself. Special techniques have to be used to link the material points between the reference framework and the current framework [99, 100].

The *Eulerian-Lagrangian* models solve the *Eulerian* form of the Navier-Stokes equations for fluid on a fixed grid using a finite volume method, e.g., computational fluid dynamics (CFD), and track the moving body (structure) in a *Lagrangian* fashion. A typical example is the CFD-FEM model [101-104]. An alternative, the *Arbitrary Lagrangian-Eulerian* method (ALE), was developed to allow arbitrary motion of grid/mesh points with respect to their frame of reference by taking the convection of these points into account. However, for large translations and rotations of the solid or inhomogeneous movements of the mesh points the fluid elements tend to become ill-shaped, which reflects on the accuracy of the solution. Remeshing, in which the whole domain or part of the domain is spatially discretised, is then a common strategy. The process of generating mesh multiple times during a computation can, however, be a very troublesome and time consuming task. In particular the contact of the elastic structure with the boundary is not possible within a monolithic formulation using simple ALE coordinates without remeshing techniques [105].

Even though some remedies have been used to minimise those limitations [106, 107], the features such as large deformation, free surfaces and deformable boundaries are still great challenges in coupled CFD-FEM models and conventional Eulerian-Eulerian methods and Euler-Lagrangian methods can only solve FSI problems where the structure immersed in the fluid field deforms without any fracture. On the other hand in the meshfree methods, the identification of free surfaces, moving interfaces and deformable boundaries can be handled straightforwardly [108]. Due to those evident advantages in meshfree methods, some research efforts have been focused on coupling meshfree methods with CFD [24] or FEM [109, 110], and even developing coupled meshfree models such as SPH-SPH [23], SPH-DEM [24, 111, 112] and LBM-DEM [109].

The presence of free surface flow in the concerned FSI problems makes SPH preferred to remain in the coupled model to be developed. Among the above models the coupled SPH-FEM model [113], the coupled SPH-SPH model [23, 114] and the coupled SPH-DEM [24, 111, 112] are *Lagrangian-Lagrangian* schemes. As both fluid and structure components are represented in the same framework, the coupling at the interface can be easily achieved. In addition, as both SPH and DEM are lagrangian particle methods, they can be computationally accelerated for large scale simulations by using GPU technique which has been already in individual SPH and DEM models. In DEM, the strength of the bond and the fracture criteria are derived through Young's modulus and Poisson's ratio which are the physical properties of real material. What is more, coupled SPH-DEM model can do not only FSI problems but also Fluid-Particle-Structure Interaction (FPSI) problems. In addition, as both SPH and DEM are

lagrangian particle methods, they can be computationally accelerated for large scale simulations by using GPU technique which has been already in individual SPH and DEM models. The coupled model applied in FSI problems is the first step before advancing to FPSI problems. These models are capable of simulating the free-surface flow and dynamic boundary problems involved in FSI problems, but the kernel functions used in SPH for solid structure lack a physical representation of fracture, not to mention further complications such as the permeation of fluid in the porous or fractured zones of solid structure and the large deformation in FEM is still under numerical challenges. In the coupled SPH-DEM models developed in [24, 111] the structures are treated as rigid bodies thus the interaction between structure and fluid is not fully studied and the deformation and fracture of structure has not been achieved. Even if the structures in FSI problems with free surface flow is represented by SPH or FEM, to the authors' best knowledge, none of those models is capable of dealing with fracture or crack initiation in the structure part during the FSI process.

The FEM as a traditional mesh-based method and its extended versions play an important role in dealing with solid fracture or structural failure problems [115, 116]. Phantom-Node method [117] was also incorporated into FEM through integration of overlapped elements in order to handle crack kinematics, but the crack-tip enrichment is still challenging and its flexibility is comprised when crack growth is the only focus. Therefore coupling FEM with SPH for modelling fluid induced structural failure during the FSI process would become even more challenging.

Another method referred as continuous/discontinuous deformation analysis (CDDA) [118] was developed to account for fracture by employing a link element to connect two adjacent elements as a virtual crack extension. Alternatively, meshfree methods [119, 120] as a promising technique in recent years have been applied in modelling of fracture. The development of test and trial function with a sign function can model cracks with arbitrary movement [26]. Rabczuk [25] used immersed particle method treated in fluid and structure, in which a Kirchhoff–Love shell theory is adopted, to model FSI with crack propagation. A cubic/quartic polynomial basis [121] was used in meshfree particle methods, but without taking the gradient of a kernel function to model cracks the polynomial functions used for solid structure lack a physical representation of fracture unlike the traditional constitutive laws described in solid mechanics. Even though these methods are promising in dealing with crack failure, their applicability is limited by extending to more complex Fluid-Particle-Structure Interaction (FPSI) system.

As another type of meshfree methods, DEM, has recently been successfully applied to model the fracture of solids such as ceramics [22], concrete [122] and even composite materials [123]. The particles in DEM are bonded together and the crack initiation and propagation is treated as the progressive breakings of bonds. The crack pattern is automatically determined without any need of re-meshing and can be dynamically visualized during the simulation process. DEM model does not require the formulation of complex constitutive laws that are essential in FEM model, while it requires calibration with measured macro-scale results to determine the micro-scale particle and contact parameters that will predict the macro-

scale response. Therefore, DEM is practical for studying general features of the statics and dynamics of fracturing, like the crack shape, global structural failure due to the collective behaviour of many interacting cracks as well as the dynamic instability of cracks during their propagation.

1.5 Motivations and objectives

Up to now, mesh-based methods have been extensively applied in the engineering problems, but due to its limitation in moving boundary problems like large deformation and free surface flow, its computational efficiency and numerical accuracy is largely affected, even though some remedies have been proposed for optimisation. Therefore, there is a demand in developing a completely new method to overcome these limitations. Upon the proposal of meshfree methods, they can be easily implemented to deal with moving boundary problems. When dealing with fluid-structure interaction, coupled meshfree methods have been developed with a huge progress in comparison with mesh-based methods. However, the structure is initially assumed to be rigid bodies in the most of current studies, even there is no research to deal with the failure of the structure. Besides fluid-structure interaction, fluid-particle interaction has been widely studied, but there is rare research in investigating fluid-particle-structure interaction.

Therefore, this thesis aims to present a new approach based on fully meshfree particle methods of SPH and DEM to handle the FSI problems with free surface flow and/or structural failure. One of the objectives of this research is to develop an advanced FSI model for investigating multi-physics problems. To address this kind of problem, interdisciplinary knowledge of geotechnical, hydraulic and structural engineering are required, and it also raises a demand on a robust and reliable computer model to predict the interaction among different phases. Thus a numerical model for fluid-solid (FSI) interaction would be extremely helpful for assessing the risk of bridge

collapse and also assisting the development of dedicated strengthening technique to prevent the failure of the bridge at risk.

According to the originalities in this study as follow:

1. A pure coupled meshfree methods,
2. Structure phase can be a non-rigid body to allow the structural deformation and failure,
3. Combination of fluid-structure interaction and fluid-particle interaction into fluid-particle-structure interaction.

The SPH-DEM model presented in this paper is the first step of developing a unified particle model for general FSI problems in engineering. The coupled SPH-DEM model will be able to capture either the deformation or the fracture events in the solid structure induced by the free surface flow of the fluid. In this approach, the SPH based on the Navier-Stokes equations is used to model the fluid domain. The DEM is used to represent the solid structure through a dense packing of bonded particles which allows deformation and/or fracture. Similar approaches have already been adopted for modelling ceramics [22] and concrete [124]. As the interaction between discrete particles can be naturally taken into account by DEM, the coupled SPH-DEM presented in this study for FSI has the potential of being easily extended to model the interaction between fluid phase, solid particle phase and structure phase separately or simultaneously, and applied to address the FSI problems in engineering.

1.6 Thesis outline

Chapter One presents a brief introduction to the background of fluid-solid interaction followed by the introduction of mesh-based methods and meshfree methods. And then two meshfree methods: Smoothed Particle Hydrodynamics (SPH) and Discrete Element Method (DEM) are reviewed in details. And then the modelling software PFC2D 5.0 is presented with the definition of the aims and objectives of the research studies proposed.

Chapter Two and Three detail a full presentation of the theory behind the Smooth Particle Hydrodynamics (SPH) for fluid phase and the Discrete Element Method (DEM) models for solid phase including, a definition of the calculation principles and a description of the relevant constitutive models employed from the PFC2D 5.0 used in the execution of this project. And chapter Four presents the techniques and physical model used in the fluid-solid interaction.

Chapter Five to Seven reports the validation results of SPH, DEM and SPH-DEM respectively. A series of validation cases have been tested to prove the proposed integrated particle model.

Chapter Eight and Nine applied the validated model to investigate two engineering problems: 3D Printing of Fibre Reinforced Polymer Composites and 2D simulation of injection moulding process of short fibre.

Chapter Ten presents a review of the major conclusions and research contributions made by this study and a discussion of the potential future extensions that may be pursued.

2 SPH for fluid dynamics

2.1 Navier-Stokes equations in SPH form

The governing equations of Navier-Stokes equations are based on the following three fundamental physical laws of conservation.

1. Conservation of mass
2. Conservation of momentum
3. Conservation of energy

Different forms of equations can be employed to describe the fluid flows, depending on the specific circumstances. As discussed, there are two approaches for describing the physical governing equations, the Eulerian description and Lagrangian description. The Eulerian description is a spatial description, whereas the Lagrangian description is a material description. The fundamental difference of these two descriptions is that the Lagrangian description employs the total time derivative as the combination of local derivative and convective derivative. The SPH equations of motion will be derived based on these governing equations in Lagrangian form. In order to transform partial derivative form into ordinary differential equations (ODEs) under SPH scheme, the following steps are presented in detail.

2.1.1 Interpolation of a function and interpolation of the derivative of a function

The formulation of SPH is made up of two key steps, kernel approximation and particle approximation. In the first step, the typical integral forms of a function is given by the multiplication of an arbitrary function and a smoothing kernel function, and its derivative are described by simply substituting $f(x)$ with $\nabla \cdot f(x)$ and finally formatted as:

$$f(x) = \int_{\Omega} f(x')W(x - x', h)dx' \quad (1)$$

$$\nabla \cdot f(x) = \int_s f(x')W(x - x', h) \cdot \vec{n} dx' - \int_{\Omega} f(x') \cdot \nabla W(x - x', h)dx' \quad (2)$$

In the second step for the particle approximation, which is another key operation in the SPH methods, as the entire SPH system is represented by a finite number of particles that carry individual mass and occupy individual space, the integral representation of the function and its derivative is approximated by summing up the values of influential surrounding particles and this step is usually called particle approximation, as shown in Fig.1. Only the particles located in the support domain of kernel function with a radius of kh are taken to account in particle approximation. As a result, the final forms of Eqs.(1) and (2) are approximated as:

$$f(x_i) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j)W_{ij} \quad (3)$$

$$\nabla \cdot f(x) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) \cdot \nabla_i W_{ij} \quad (4)$$

where i and j in subscript denote particle i and j , N is the number of particles within the support domain of the kernel function, m is the mass of the particle and ρ is the density of the particle.

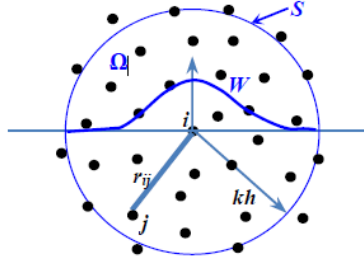


Fig. 1 Particle approximations for particle i within the support domain kh of the kernel function W . r_{ij} is the distance between particle i and j , s is the surface of integration domain, Ω is the circular integration domain, k is the constant related to kernel function and h is the smooth length of kernel function.

For continuity equation, the rate of change of density and momentum equation, the conservation of momentum, in Navier-Stokes form are given by:

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot v \quad (5)$$

$$\frac{Dv}{Dt} = -\frac{1}{\rho} \frac{\partial \sigma}{\partial x} \quad (6)$$

where ρ and $\nabla \cdot v$ are mass and velocity divergence respectively, σ is the total stress tensor. It is made up of two parts, one part of isotropic pressure P and the other part of viscous stress τ .

As energy equation is not considered in the current research, it is not presented here, but it can be referred to [108]

In SPH form, the continuity equation Eq. (5) becomes:

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^N m_j v_{ij} \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (7)$$

In addition, the density of particle can be directly calculated by summing up all the particles' mass together since the integration of density over the entire problem domain is exactly the total mass of all the particles:

$$\rho_i = \sum_{j=1}^N m_j W_{ij} \quad (8)$$

However, the summation density approach is influenced by the boundaries where the domain of the kernel function is partly truncated, and the non-zero surface integral is directly the result of truncation. One of the accuracy improvements has been proposed to normalise Eq. (8) by summing up the kernel function over the surrounding particles [125]:

$$\rho_i = \frac{\sum_{j=1}^N m_j W_{ij}}{\sum_{j=1}^N \left(\frac{m_j}{\rho_j}\right) W_{ij}} \quad (9)$$

In concern with the discontinuity at boundary or interface, the density integrated by $\frac{D\rho_i}{Dt}$ can assure the preservation of discontinuity all the time with much less computational cost. Therefore, continuity density approach is the default one to calculate particle density.

The derivation of SPH formulations for particle approximation of momentum equation is similar to the continuity density approach, and usually involves some transformations. In SPH form, the momentum equation Eq. (6) becomes:

$$\frac{dv_i}{dt} = \sum_{j=1}^n m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla W_{ij} \quad (10)$$

It should be noted that neglecting the viscous term in the Navier-Stokes equation yields the Euler equation. Therefore, Eq. (6) neglects the viscosity approximation and only retaining the pressure approximations. The viscosity approximation will be discussed in the later section.

2.2 Numerical implementation of SPH

In this study, the SPH theory above is implemented in PFC2D v5.0 using C++. The indirect contact feature is adopted to enable the particle interaction in SPH. PFC2D 5.0 as a DEM software package which has many features that can be directly utilised for SPH simulations such as a particle search scheme and a time integration scheme. A particle search scheme is based on a Linked-list algorithm to sub-divide the particles within different cells and particles are identified through a linked list. PFC2D 5.0 uses a leapfrog technique for numerical integration to update field variables at each particle. As the codes are written in C++, they are portable for other open source DEM codes for SPH-DEM simulations without much modification.

2.2.1 Kernel selections

Up to now, various kernel functions have been developed and used in the SPH method [108], among which the most widely used are the cubic spline kernel function [126] and the Wendland kernel function [127].

$$\text{Cubic spline} \quad -W(r, h) = C_h \begin{cases} (2 - q)^3 - 4(1 - q)^3 & \text{for } 0 \leq q \leq 1 \\ (2 - q)^3 & \text{for } 1 < q \leq 2 \\ 0 & \text{for } q > 2 \end{cases} \quad (11)$$

where $q = |r|/h$. $|r|$ is the distance between two particles, h is the smoothing kernel length associated with a particle and the normalisation is ensured by setting up the constant C_h to be $15/(14\pi h^2)$ in two dimensions.

$$\text{Wendland} \quad W(r, h) = C_h \begin{cases} (2 - q)^4(1 + 2q) & \text{for } 0 \leq q \leq 2 \\ 0 & \text{for } q > 2 \end{cases} \quad (12)$$

where C_h in two dimensions is normalised to be $7/(64\pi h^2)$

Static tank tests are carried out using SPH with both kernel functions. According to the results shown in later section, the simulation using Wendland kernel show more orderly distribution of particle than cubic spline kernel, as a result, the Wendland kernel is chosen for all simulations in this study.

2.2.2 Physical Viscosity

There is a wide variety of derivation of the viscosity term [128], and the first one derived as an artificial viscosity is based on the consideration of strong shocks [129]:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c \bar{\mu}_{ij} + \beta \bar{\mu}_{ij}^2}{\bar{\rho}_{ij}} & \text{if } v_{ij} \cdot r_{ij} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where α and β denotes the artificial viscosity coefficient respectively, $\bar{\mu}_{ij} = 1/2(\mu_i + \mu_j)$ and $\bar{\rho}_{ij} = 1/2(\rho_i + \rho_j)$. As it has been a common practice to use an artificial viscosity in compressible SPH formulations for better accuracy in the simulation of shock wave, this viscosity form will not be taken into consideration here. Instead, another viscosity form including physical viscosity of particle derived in [130] is adopted in this study:

$$\Pi_{ij} = m_j \frac{(\mu_i + \mu_j) r_{ij} \cdot \nabla W_{ij}}{\rho_i \rho_j (r_{ij}^2 + 0.01h^2)} v_{ij} \quad (14)$$

where $0.01h^2$ in the denominator is meant to avoid singularity.

Therefore the momentum in SPH form is updated to:

$$\frac{dv_i}{dt} = \sum_{j=1}^n m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij} \quad (15)$$

Apparently Eq.(14) can approximate the viscosity term physically and it is also useful for dealing with multiphase problems where densities at interface are not identical. This will become more important when discrete particles are incorporated in the present SPH-DEM model in the future to enable the FSI simulations.

2.2.3 Tensile instability

Even though SPH is an increasingly promising numerical method, several difficulties have been encountered in recent decades. The first difficulty is the completeness of SPH which is the ability of the approximation to reproduce specified functions and another ones are the rank deficiency and the tensile instability that manifests itself as a bunching of nodes. This unphysical phenomenon, which is normally due to tensile instability, could reduce resolution and even cause numerical errors during the simulation. To extend applications of SPH into a wide range of fluid dynamics problems, scholars have conducted a series of modifications and corrections to improve the approximation accuracy.

In terms of completeness, there are two approaches to approximate continuity equation: one is density summation approach and another one is

continuity density approach. The density summation approach conserves the mass since the integration of density over the entire support domain is equal to the total mass of all the particles. However, this approach suffered edge effect, namely boundary particle deficiency where the support domain is not fully filled with particle at the edge of fluid domain, as a result, the density is smoothed out to cause some spurious results. Randles and Libersky [59] proposed the normalisation of the summation density approach with the SPH summation of the smoothing function itself over the surrounding particles to improve the accuracy of approximation. In this study, the continuity density approach was applied instead of the density summation approach to introduce velocity difference into the discrete particle approximation as the usage of the relative velocities in anti-symmetrized form serves to reduce errors arising from the particle inconsistency problem [108].

The rank-deficiency was defined as that the number of integration points is less enough so that the solution to the underlying equilibrium equation becomes non-unique. Even though scholars [62, 131] proposed to eliminate the rank-deficiency by introducing additional integration points (e.g. stress points) at other locations than the SPH centroids, the increased computational effort associated with the additional integration points renders this approach less efficient, and a precise guideline as to how many additional stress points are needed is missing. In this study, several smooth lengths were tested to find out the optimal value of smooth length in order to keep the rank deficiency as minimum as possible.

The original updated Lagrangian formulation of SPH, which is also termed Eulerian SPH, suffers from the so-called tensile instability, in which leads to the clumping of particles. A Lagrangian formulation where the kernel approximation is performed in the initial, undeformed reference coordinates of the material [26]. In the application of Lagrangian kernel, the tensile instability is absent, however, rank deficiency is still existed. In this study, the instability can be removed by using an artificial stress which, in the case of fluids, is an artificial pressure. An anti-clump term was introduced to be added into the momentum equations to prevent particles from forming into small clumps due to unwanted attraction [45]:

$$\frac{dv_i}{dt} = \sum_{j=1}^n m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} + R_{ij} \right) \nabla W_{ij} \quad (16)$$

$$R_{ij} = \frac{v_{max}^2}{c_s^2} \left| \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right| \left(\frac{W_{ij}}{W_{(\Delta P)}} \right)^4 \quad (17)$$

where $v_{max} = \frac{1}{10} c_s$, and ΔP is the initial particle spacing.

Another remedy also applied in this model is to correct the rate of the change of particle position in order to keep particles move orderly in the high speed flow:

$$\frac{dr_i}{dt} = v_i + \varepsilon \sum_j \frac{m_j (v_j - v_i)}{\bar{\rho}_{ij}} W_{ij} \quad (18)$$

where the second term on the right hand side of Eq. (18) is the correction factor, the value of ε is problem-dependent as large ε can slow down the particle velocity unphysically

2.2.4 Boundary Treatment

When a SPH particle is approaching a boundary (see Fig.2), its support domain overlaps with the problem domain, consequently its kernel function is truncated partially by the boundary and the surface integral is no longer zero. Theoretically only particles located inside the support domain are accounted for in the summation of the particle interaction, but there are no particles existing in the truncated area beyond the solid boundary. Different remedies have been proposed recently to rectify boundary truncation. The normalisation formulation of density approximation was derived to satisfy the normalisation condition and ensures the integral of kernel function over the support domain is unity [125]. In comparison with kernel re-normalisation, the application of virtual or ghost particles is widely used to replace the solid boundary and to produce a repulsive force in order to avoid wall penetration [132]. The interaction force between a boundary particle and an SPH particle could be in Lennard-Jones form [132], in which the SPH particles are repelled within a cut-off distance, but Lennard-Jones form is highly dependent on the problem being simulated. In order to have a simple, robust as well as reliable, interaction between boundary and SPH particles, in this study two-layers of fixed boundary particles are placed as solid boundaries, which are initialised with a reference density of SPH particles, but their density and other parameters such as position and velocity are all fixed and not evolved with the parameter variance of SPH particles. In order to produce sufficient repulsive forces, the distribution of boundary particles is denser than the distribution of SPH particles as shown in Fig.3.

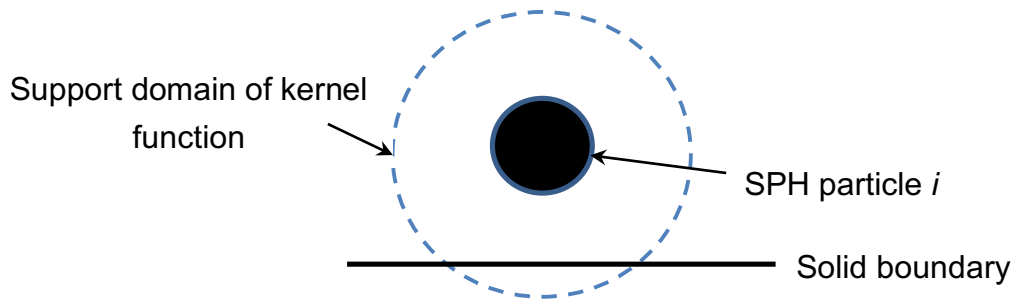


Fig. 2 Truncation of particle support domain by a boundary

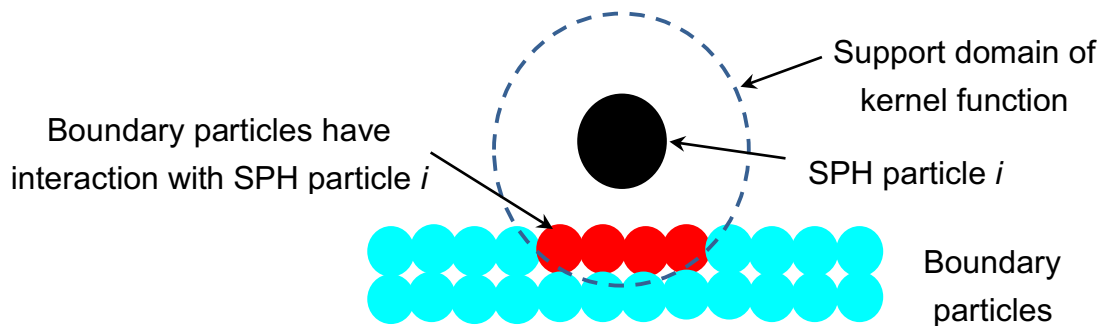


Fig. 3 Boundary particles and their interaction with SPH particles

2.2.7 Time Integration

The leapfrog time integration scheme is frequently used in particle simulation algorithms because

1. It is under explicit condition, which makes it easy to implement.
2. Second-order accuracy is guaranteed.

The leapfrog time integration algorithm is named because the velocities are updated on half steps and the positions on integer steps; hence, the two leap over each other. After computing accelerations, one step takes the form

$$v^{i+1/2} = v^{i-1/2} + a^i \Delta t \quad (19)$$

$$r^{i+1} = r^i + v^{i+1/2} \Delta t \quad (20)$$

This is straightforward enough, but it should be noted that in order to compute the acceleration at time t , the velocity at time t is needed. The leapfrog time integration only computes velocities at half steps.

3 DEM for granular flow and structure mechanism

3.1 Principles of DEM

3.1.1 Force-displacement law

The force-displacement law for a contact between two particles without the presence of a bond is usually based on contact mechanics theory such as Hertz linear contact theory [133]. When a bond is assigned to the contact, the overall force-displacement of the bonded particles is a combination of particle and bond properties. As fracture of bonds, which could induce pure particle-particle contact on the cracked surfaces, will be allowed in some of the simulations presented in this paper, the force-displacement law for pure particle-particle contact is briefly described first, followed by the constitutive law of the bond. More details are available in the literature [134, 135].

At the contact between two unbonded particles, the contact force vector is further resolved into normal and shear components with respect to the contact plane (see Fig.4) as follows:

$$F = F^n + F^s \quad (21)$$

where F^n and F^s denote the normal and shear components, respectively.

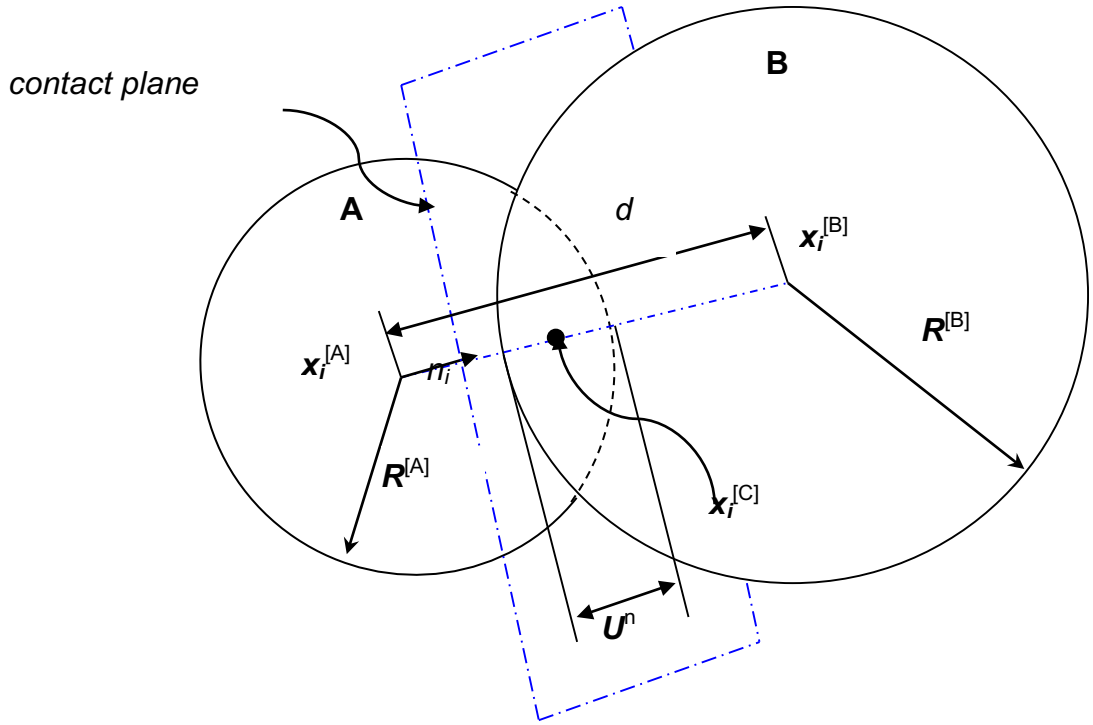


Fig. 4 Two particles in direct contact

The magnitude of the normal force is the product of the normal stiffness at the contact and the overlap between the two particles, *i.e.*,

$$F^n = K^n U^n \quad (22)$$

where K^n is the normal stiffness and U^n is the overlap.

The shear force is calculated in an incremental fashion. Initially the total shear force is set to zero upon the formation of contact and then in each timestep the relative incremental shear-displacement is added to the previous value in last timestep:

$$F^s = F^s + \Delta F^s \quad (23)$$

$$\Delta F^s = -K^s \Delta U^s \quad (24)$$

$$\Delta U^s = V^s \Delta t \quad (25)$$

where K^s is the shear stiffness at the contact, ΔU^s is the shear component of the contact displacement, V^s is the shear component of the contact velocity and Δt is the timestep.

In addition, the maximum allowable shear contact force is limited by the slip condition:

$$F_{max}^s = \mu |F^n| \quad (26)$$

where μ is the friction coefficient at the contact.

In cases where a steady-state solution is required in a reasonable number of cycles, the dashpot force acting as viscous damping is grouped into the force-displacement law to account for the compensation of insufficient frictional sliding or no frictional sliding. In line with spring forces, the dashpot force is also resolved into normal and shear components at the contact:

$$F_n^d = 2\beta_n \sqrt{mK_n} \delta_n \quad (27)$$

$$F_s^d = 2\beta_s \sqrt{mK_s} \delta_s \quad (28)$$

$$m = \frac{m_A m_B}{m_A + m_B} \quad (29)$$

where d in superscript denotes dashpot, A and B in subscript denote the two particles in the contact pair, β is the critical damping ratio and δ is the relative velocity difference between two particles in contact.

When a bond is created between two particles, the normal and shear components of the bond force are included in the force-displacement law. It is noted that normal bond force is first examined to see if the tensile-strength limit is exceeded. If a bond is still present in the tension state, the shear-

strength limit is enforced for second iteration. When the bond is broken the bond force is diminished in the force-displacement law. Details of the fracture of bonds in DEM will be discussed in later section.

3.1.2 Law of motion

The motion of each particle in each timestep is governed by Newton's Second Law in terms of translational and rotational motions as follow

Translational

$$F_i = m(\ddot{x}_i - g_i) \quad (30)$$

motion:

Rotational motion:

$$M_i = I\dot{\omega}_i \quad (31)$$

where i in subscript is the indicial notation with respect to coordinate system, F_i is the resultant force, m is the total mass of particle, g_i is the body force acceleration vector, \ddot{x}_i is the acceleration vector of a particle, M_i is the resultant moment acting on a particle, I is the principal moment of inertia of the particle, and $\dot{\omega}_i$ is the angular acceleration about the principal axes.

The leapfrog method is used to update the position of the particle. First the relationship between the acceleration and velocity is defined by

$$\ddot{x}_i^{(t)} = \frac{1}{\Delta t} (\dot{x}_i^{(t+\frac{\Delta t}{2})} - \dot{x}_i^{(t-\frac{\Delta t}{2})}) \quad (32)$$

$$\dot{\omega}_i^{(t)} = \frac{1}{\Delta t} (\omega_i^{(t+\frac{\Delta t}{2})} - \omega_i^{(t-\frac{\Delta t}{2})}) \quad (33)$$

Then Eqs. (32) and (33) are substituted into Eqs. (30) and (31) respectively and the velocity at time $(t + \frac{\Delta t}{2})$ is resolved as:

$$\dot{x}_i^{(t+\frac{\Delta t}{2})} = \dot{x}_i^{(t-\frac{\Delta t}{2})} + \left(\frac{F_i^{(t)}}{m} + g_i\right)\Delta t \quad (34)$$

$$\omega_i^{(t+\frac{\Delta t}{2})} = \omega_i^{(t-\frac{\Delta t}{2})} + \left(\frac{M_i^{(t)}}{I}\right)\Delta t \quad (35)$$

Finally the position of the particle is updated accordingly:

$$x_i^{(t+\Delta t)} = x_i^{(t)} + \dot{x}_i^{(t+\frac{\Delta t}{2})}\Delta t \quad (36)$$

3.2 Contact models

3.2.1 Linear contact model

Linear contact model is widely used throughout the DEM field and was first published in [98]. The linear model shown in Fig.5 provides linear and dashpot components that act in parallel with one another. The linear component provides linear elastic (no-tension), frictional behaviour, while the dashpot component provides viscous behaviour. The linear force is produced by linear springs with constant normal stiffness K^n and shear stiffness K^s . The dashpot force is produced by dashpots with viscosity given in terms of the normal and shear critical-damping ratios. The linear springs act in parallel with the dashpots. A surface gap is defined as the difference between the contact gap and the reference gap so that when the reference gap is zero. The contact is active if the surface gap is less than or equal to zero; the force-displacement law is skipped for inactive contacts. Both components act over a vanishingly small area, and thus transmit only a force. Normal force is linear function of normal displacement (=overlap);

shear force increases linearly with relative shear displacement, but is limited by Coulomb linear friction.

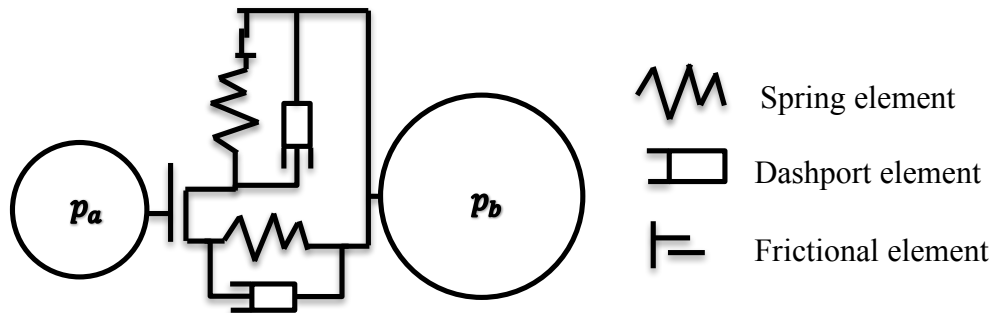


Fig. 5 2D representation of a contact between two particle elements in DEM

3.2.2 Linear parallel bond model

The linear parallel bond model with inactive dashpots and a reference gap of zero corresponds with the parallel-bond model of [135]. Fig.6 shows that a parallel bond is a finite-sized piece of cement-like material deposited between the two contacting pieces. The parallel-bond component acts in parallel with the linear component and establishes an elastic interaction between the pieces. The existence of a parallel bond does not preclude the possibility of slip. Parallel bonds can transmit both force and moment between the pieces.

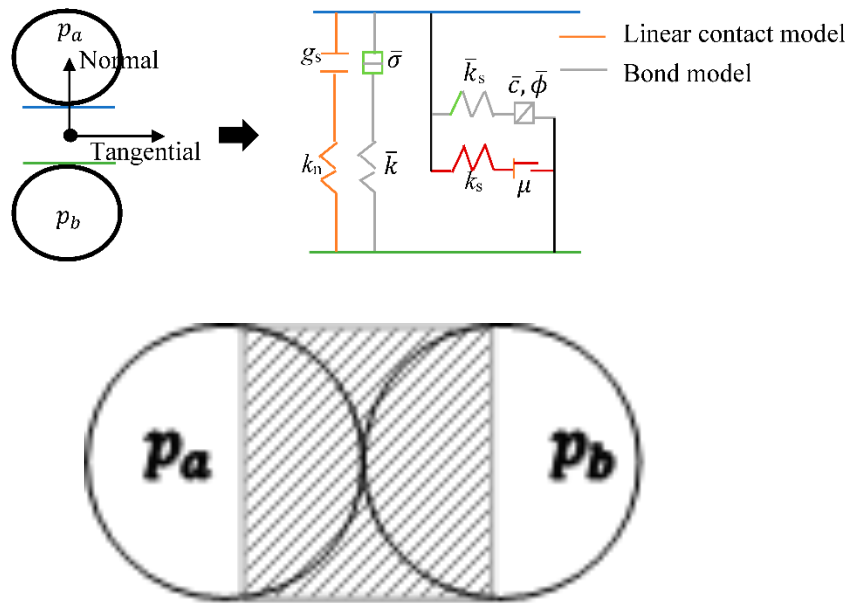


Fig. 6 DEM particle elements with a parallel bond

The linear parallel bond model provides the behaviour of two interfaces: an infinitesimal, linear elastic (no-tension) and frictional interface that carries a force and a finite-size, linear elastic and bonded interface that carries a force and moment. The first interface is equivalent to the linear model: it does not resist relative rotation, and slip is accommodated by imposing a Coulomb limit on the shear force. The second interface is called a parallel bond, because when bonded, it acts in parallel with the first interface. When the second interface is bonded, it resists relative rotation, and its behaviour is linear elastic until the strength limit is exceeded and the bond breaks making it unbonded. When the second interface is unbonded, it carries no load. The unbonded linear parallel bond model is equivalent to the linear model

3.3 Fracture criteria of structure

3.3.1 Particle distribution

Particles can be packed in a regular (e.g. hexagonal or cubic in 2D) or random form. When they are packed in a hexagonal form in plane stress condition, as shown in Fig.7, the relationship between the elasticity of the constructed structure and the stiffness of the contacts can be derived as [136]:

$$K_n = \frac{E\delta}{\sqrt{3}(1-\nu)} \quad (36)$$

$$K_s = \frac{E\delta(1-3\nu)}{\sqrt{3}(1-\nu^2)} \quad (37)$$

where K_n and K_s are the contact stiffness in normal and shear directions respectively, E is the Young's modulus, δ is the element thickness and ν is the Poisson's ratio.

As illustrated in Eq. (37), there is a constraint of $(1-3\nu)$ term on the right-hand side of equation in which the value of Poisson's ratio needs to be smaller than or equal to 0.33 so as to guarantee a positive value of K_s .

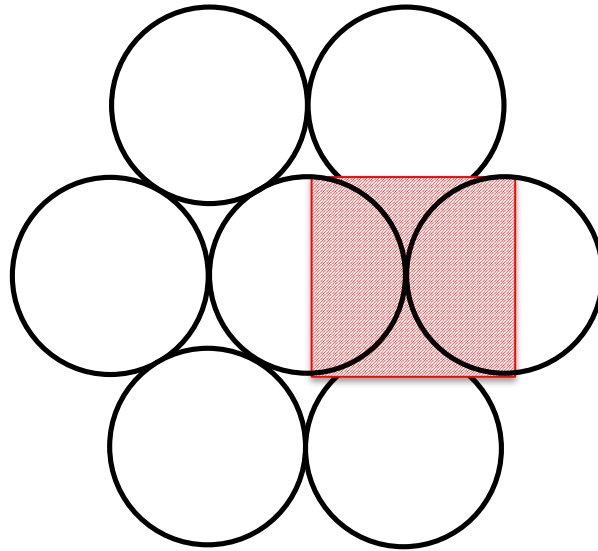


Fig. 7 Hexagonal packing of discrete particles with parallel bonds

3.3.2 Bond stiffness determination

A bond in DEM can be regarded as a glue to stick two particles together, and linear parallel bond is special bond (see rectangular box indicated in Fig.7) that can be decomposed into linear model and parallel bond model which are acting in parallel. The bond is broken when the strength limit of bond is exceeded [123, 134] and after that only the linear model is active. Upon the use of a linear parallel bond model, the contact stiffness K_i is the result of combination of both particles' stiffness and bond stiffness according to the following formulation [134]:

$$K_i = A\bar{k}_i + k_i \quad (38)$$

$$A = 2\bar{R}\delta \quad (39)$$

$$k_i = \frac{k_i^{[A]}k_i^{[B]}}{k_i^{[A]} + k_i^{[B]}} \quad (40)$$

Where \bar{R} and A are the radius and cross-sectional area of the bond, respectively, \bar{k}_i is the parallel bond stiffness and k_i is the equivalent stiffness

of two contacting particles. In this study the radius of the bond is the same as the particle radius. If two particles have the same normal and shear stiffness, k_i is simplified as:

$$k_i = \frac{k_i^{[A]}}{2} = \frac{k_i^{[B]}}{2} \quad (41)$$

It can be assumed that the parallel bond stiffness is much larger than the particles' stiffness, thus the forces are predominantly passed through parallel bonds, *i.e.* $k_i = 0.01A\bar{k}_i$,

$$K_i \approx A\bar{k}_i \quad (42)$$

Thus the parallel bond stiffness is determined by combining Eqs. (38) and (39) with Eq.(42).

3.3.3 Bond ultimate strength determination

According to the nature of parallel bond model, bond strength is the only criterion to determine the fracture of a structure. When the structure is under pure tension, the bond strength can be derived in terms of ultimate tensile strength and Poisson's ratio [124]:

$$f_n^{crit} = \frac{\bar{R}\delta\sigma_{ult}}{2(1-\nu)} \left(\sqrt{3} - \frac{\nu}{\sqrt{3}} \right) \quad (43)$$

$$f_s^{crit} = \frac{\bar{R}\delta\sigma_{ult}}{2(1-\nu)} (1 - 3\nu) \quad (44)$$

$$\sigma_n^{crit} = \frac{f_n^{crit}}{2\bar{R}\delta} \quad (45)$$

$$\sigma_s^{crit} = \frac{f_s^{crit}}{2\bar{R}\delta} \quad (46)$$

where f_n^{crit} and f_s^{crit} are maximum normal and shear forces acting on the parallel bond, σ_n^{crit} and σ_s^{crit} are critical tensile and shear stresses. It should be noted that the above derivation is only valid for 2D simulations in plane stress condition.

During the simulation, the parallel bond forces in normal and shear directions are updated at each timestep through the force-displacement law:

$$f_n = A\bar{k}_n\Delta\delta_n \quad (47)$$

$$f_s = -A\bar{k}_s\Delta\delta_s \quad (48)$$

$$\sigma_n = \frac{f_n}{A} + \bar{\beta}\frac{M_b\bar{R}}{I} = \bar{k}_n\Delta\delta_n + \bar{\beta}\frac{M_b\bar{R}}{I} \quad (49)$$

$$\sigma_s = \frac{|f_s|}{A} + \begin{cases} 0, (2D) \\ \bar{\beta}\frac{M_t\bar{R}}{I}, (3D) \end{cases} = \bar{k}_s\Delta\delta_s + \begin{cases} 0, (2D) \\ \bar{\beta}\frac{M_t\bar{R}}{I}, (3D) \end{cases} \quad (50)$$

where $\Delta\delta_n$ and $\Delta\delta_s$ are the relative normal-displacement increment and the relative shear-displacement increment respectively, M_b is the bending moment, M_t is the twisting moment and $\bar{\beta}$ is the moment-contribution factor. It should be noted that $\bar{\beta}$ in Eqs. (49) and (50) is set to be zero in order to match those derived formulations in Eqs. (45) and (46).

Then the strength limit is enforced to examine if the gained stresses exceed the threshold value of critical stresses. If the tensile-strength limit is exceeded (*i.e.* $\sigma_n \geq \sigma_n^{crit}$), then the bond is broken in tension, otherwise, shear-strength limit is enforced subsequently and the bond is broken in shear if $\sigma_s \geq \sigma_s^{crit}$. Once two particles are in unbonded state, parallel bond model is not active any more, but the linear particle-particle contact model, is

then activated to account for the collision of particles. More details about parallel bond can be found in [134, 135].

As seen from Eqs. (49) and (50), parallel bond is behaved linearly and the plastic deformation is not taken into consideration here. As for plastic or adhesive materials, several alternative models may be used by considering more complicated constitutive behaviour. One of them is the contact softening model [136] which is a bilinear elastic model and is similar to cohesive zone model (CZM) in continuum mechanics. In this study the structure is considered as elastic.

4 Coupling schemes for fluid-solid interaction

The model proposed in this paper is essentially dependent on the definition of interaction forces existing among the solid particles, fluid and structure(s). When considering interaction forces amongst two identical phases (e.g. fluid-fluid, solid particle-solid particle, structure-structure), it is straightforward to handle them in either SPH or DEM scheme. To avoid confusion, 'solid particle' and 'particle element' are used thereafter to distinguish a real particle (which although is represented by a particle element in DEM) and a particle element in DEM or SPH. For interaction between a solid particle and fluid, hydrodynamic force is the only force transferred to the surrounding fluid which is represented by SPH particle elements. When a pair of solid particles are in contact, the overlap and friction determine the amount of contact force. The interaction between particle elements in a structure is dominated by the addition of a bond as a glue to stick the particle elements together and represent the material properties of a structure. However, more forces should be taken into consideration for interactions between two different phases. When solid particles are fully or partially immersed within a fluid, drag force and buoyancy force from fluid particle elements physically act on the solid particles and the interaction forces between the solid particles include direct contact force as well as lubrication force due to the wet surfaces around the solid particles. By following Newton's Third law, the drag and buoyancy forces will be returned to fluid particles in equal amount but in opposite directions. As the structure is inherently built with bonded particle elements, the interaction between a particle element of the structure

and a solid particle (which is actually represented by single particle element in this study) is naturally the same as the interaction between two solid particles. The interaction between particle elements of the fluid and structure are simplified by introducing particle elements of the structure into the SPH computation algorithm to hydrodynamically interact with the particle elements of the fluid. An illustration of the integrated particle model is shown as below in Fig.8. Formulation and implementation of these interaction forces will be explained in detail in the next section along with a brief introduction of SPH and DEM theories.

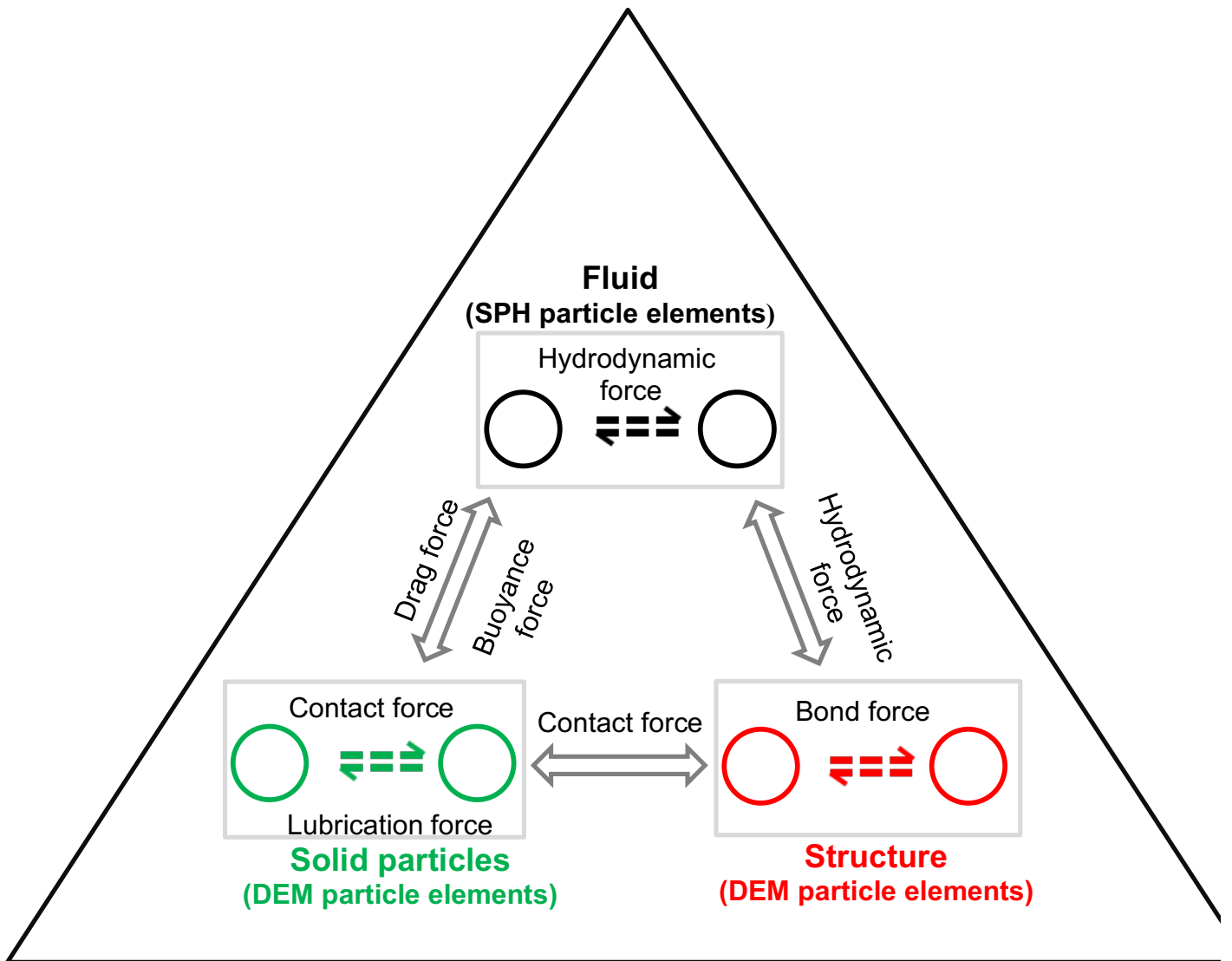


Fig. 8 Schematic diagram of interaction forces in the integrated particle model

4.1 Interaction forces for fluid particle in SPH

4.1.1 Local averaging technique

When dealing with a large amount of closely packed particles suspended within the fluid, it is too complicated to obtain direct solutions of the Navier-Stokes equations and the Newtonian equations of motion. Therefore, Anderson and Jackson [137] established a local averaging technique to replace mechanical variables (e.g. fluid density, fluid velocity or velocity of solid matters) by defining local mean variables over fluid regions or solid regions, which are smoothed out by a radial smoothing function.

The local average of any field \acute{a} over a fluid domain can be derived by the convolution with the smoothing function as follow:

$$\epsilon(x_1)a(x_1) = \int_{v_f} \acute{a}(x_2)g(x_1 - x_2)dV \quad (51)$$

$$\epsilon(x_1) = 1 - \int_{v_p} g(x_1 - x_2)dV \quad (52)$$

Where x_1 and x_2 are coordinates of position and one dimension is assumed here for simplicity, ϵ is the local mean voidage, g is the smoothing function and v_f and v_p are volumes of fluid and solid particle, respectively. The integral is taken over the volumes of fluid or solid particle.

In a similar fashion, the local average of any field a over solid domain can be derived by integrating over the volume of solid particles:

$$(1 - \epsilon(x_1))a(x_1) = \int_{v_s} \acute{a}(x_2)g(x_1 - x_2)dV \quad (53)$$

where the integral is taken over the volume of solid particle.

As the local volume fraction of fluid phase is mathematically important to define the spatial distribution of phase density, the locally averaged fluid density $\bar{\rho}_f$ is then the product of the actual fluid density ρ_f and the local mean voidage of fluid ϵ :

$$\bar{\rho}_f = \epsilon \times \rho_f \quad (54)$$

The derived locally averaged fluid density is subsequently applied in the Navier-Stokes equations without considering the energy equation of the fluid phase and it is written as:

$$\frac{D\bar{\rho}_f}{Dt} + \bar{\rho}_f \nabla \cdot v_f = 0 \quad (55)$$

$$\bar{\rho}_f \frac{Dv_f}{Dt} = -\epsilon \nabla p - F_f^d - F_f^{fs} + \nabla \cdot \tau + \bar{\rho}_f g \quad (56)$$

where v_f is the fluid velocity, p is the fluid pressure, F_f^d is the fluid-particle interaction force per unit volume acting on fluid 'particles' due to drag force acting on solid particles, F_f^{fs} is the fluid-structure interaction force per unit volume, and τ and g stand for the stress deviator tensor and gravitational acceleration, respectively.

The motion of each solid particle is governed by various forces (e.g. drag force, lubrication force due to wet surfaces between particle pair and buoyancy force) which can be taken into consideration as follows:

$$m_p \frac{dv_p}{dt} = \sum F_p^c + \sum F_p^l + m_p g + F_p^d + F_p^b + F_p^{ps} \quad (57)$$

where subscript p in this study is used to define the solid particle, v_p is the velocity of solid particle, F_p^c is the sum of direct contact forces between the solid particles. F_p^l is the sum of lubrication forces arising between particles

immersed in the fluid phase, m is the mass of solid particle and it vanishes in x direction, F_p^d is the drag force acting on solid particle from surrounding fluid 'particles', F_p^b is the buoyancy force and F_p^{ps} is the particle-structure interaction force.

The structure is constructed through densely packed particle elements connected by bonds which represent the material property of the structure. More details of the bonds will be given in a later section. The forces acting on the structure are primarily the internal forces arising from interparticle bonds and the external forces from fluid and solid particles:

$$m_s \frac{dv_s}{dt} = \sum F_s^b + m_s g + \sum (F_s^{fs} + F_s^{ps}), \quad (58)$$

where subscript s stands for structure, F_s^b is the sum of force transferred among bonds, m_s is the mass of a single particle element in the structure and it vanishes in x direction, and F_s^{fs} and F_s^{ps} are fluid-structure interaction force and particle-structure interaction force, respectively.

4.1 Interaction forces for solid particle in DEM

The contact force acting on a solid particle is due to its contact with other solid particles and/or the particle elements of a structure. It is computed using force-displacement law and law of motion in DEM theory.

The contact force vector at the contact is further resolved into normal and shear components with respect to the contact plane [134]:

$$F = F^{normal} + F^{shear} \quad (59)$$

where F^{normal} and F^{shear} denote the normal and shear components, respectively.

The magnitude of the normal force is the product of the normal stiffness at the contact and the overlap between the two particle elements, *i.e.*,

$$F^{normal} = K^{normal} U^{shear} \quad (60)$$

where K^{normal} is the normal stiffness and U^{normal} is the overlap.

The shear force is calculated in an incremental fashion. Initially, the total shear force is set to zero upon the formation of contact and then in each timestep, the relative incremental shear-displacement is added to the previous value in the last time step:

$$F^{shear} = F^{shear} + \Delta F^{shear} \quad (61)$$

$$\Delta F^{shear} = -K^{shear} \Delta U^{shear} \quad (62)$$

$$\Delta U^{shear} = V^{shear} \Delta t \quad (63)$$

where K^{shear} is the shear stiffness at the contact, ΔU^{shear} is the shear component of the contact displacement, V^{shear} is the shear component of the contact velocity and Δt is the timestep.

In addition, the maximum allowable shear contact force is limited by the slip condition:

$$F_{max}^{shear} = \mu |F^{normal}| \quad (64)$$

where μ is the friction coefficient at the contact.

In cases where a steady-state solution is required in a reasonable number of cycles, the dashpot force acting as viscous damping is grouped into the force-displacement law to account for the compensation of insufficient

frictional sliding or no frictional sliding. In line with spring forces, the dashpot force is also resolved into normal and shear components at the contact:

$$F^{normal,dash} = 2\beta^{normal}\sqrt{mK^{normal}}\delta^{normal} \quad (65)$$

$$F^{shear,dash} = 2\beta^{normal}\sqrt{mK^{shear}}\delta^{normal} \quad (66)$$

$$m = \frac{m_i m_j}{m_i + m_j} \quad (67)$$

where *dash* in superscript denotes dashpot, *i* and *j* in subscript denote the two particle elements in the contact pair, β is the critical damping ratio and δ is the relative velocity difference between two particle elements in contact.

The drag force acting on solid particles arises due to the resistance provided by the surrounding fluid which is represented by SPH particle elements. It mainly depends on both the relative fluid flow velocity and the local density of neighbour solid particles. The local density is derived through the local mean voidage of fluid SPH particle element, ϵ , which smooths out the nearby values of fluid SPH particle elements [138]:

$$\epsilon_p = \frac{\sum \epsilon_f V_f W_{pf}}{\sum V_f W_{pf}}, \quad (68)$$

where V_f is the volume associated to the fluid particles, W_{pf} is the kernel function used in SPH approximation, which is denoted by $W_{pf} = W(r_p - r_f, h)$, where r is the position vector and h is the smoothing length.

The drag force is formulated as follows [138]:

$$F_p^b = \frac{\beta_p}{1 - \epsilon_p} (\bar{v}_f - v_p) V_p \quad (69)$$

where β_p is the interphase momentum transfer coefficient, \bar{v}_f is the average fluid flow velocity around solid particle p .

In accordance with the threshold value of ϵ_p , the value of β_p is divided into two regimes by combining equations of Ergun [139] and Wen and Yu [140]:

$$\beta_p = \begin{cases} 150 \frac{(1 - \epsilon_p)^2 \mu_f}{\epsilon_p d_p^2} + 1.75(1 - \epsilon_p) \frac{\rho_f}{d_p} |\bar{v}_f - v_p| & \epsilon_p \leq 0.8 \\ 7.5 C_d \frac{\epsilon_p (1 - \epsilon_p)}{d_p} \rho_f |\bar{v}_f - v_p| \epsilon_p^{-2.65} & \epsilon_p > 0.8 \end{cases} \quad (70)$$

where μ_f is the viscosity of fluid, ρ_f is the reference density of fluid, C_d is the drag coefficient of a single solid particle and d_p is the diameter of solid particle.

The velocity of surrounding fluid flow is approximated using Shepard filter:

$$|\bar{v}_f| = \frac{\sum v_f V_f W_{pf}}{\sum V_f W_{pf}} \quad (71)$$

where v_f is the velocity of fluid particle.

The drag coefficient C_d is relevant to Reynolds number and given by:

$$C_d = \begin{cases} \frac{24}{Re_p} (1 + 0.15 Re_f^{0.687}) & Re_p \leq 1000 \\ 0.44 & Re_f > 1000 \end{cases} \quad (72)$$

The Reynolds number of a fluid 'particle' is formulated as follow:

$$Re_f = \frac{|\bar{v}_f - v_p| \epsilon_p \rho_f d_p}{\mu_f} \quad (73)$$

The buoyancy force generated by density differences is given by the following formula:

$$F_p^b = \epsilon_p \rho_f V_p \cdot k \quad (75)$$

where k is the unit vector parallel to the direction of the gravitational force acting on the solid particle.

When solid particles are immersed within the fluid, the surfaces of particles become wet and the friction between wet surfaces are reduced in comparison to dry surfaces. The formula of lubrication force between two wet solid particles is derived from [141] as follows:

$$F_p^l = \begin{cases} -\frac{3\pi\mu_f d_{ij}^2}{8(|x_{ij}| - d_{ij})} \frac{v_{ij} \cdot x_{ij}}{x_{ij}^2} x_{ij} & x_{ij} \leq 2d_{ij} \\ 0 & x_{ij} > 2d_{ij} \end{cases} \quad (76)$$

where i and j stand for solid particle i and solid particle j , $2d_{ij} = (d_i + d_j)/2$ is the cut-off distance and d_i/d_j is the diameter of solid particles, $v_{ij} = v_i - v_j$ and $x_{ij} = x_i - x_j$.

4.2 Interaction forces for fluid in SPH

Using local averaging technique and SPH approximations, the continuity and momentum equations in Eqs (7) and (16) can be expressed as follow:

$$\frac{D\epsilon_i \rho_i}{Dt} = \sum_{j=1}^N m_j v_{ij} \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (77)$$

$$\frac{dv_i}{dt} = - \sum_{j=1}^n m_j \left(\frac{P_i}{(\epsilon_i \rho_i)^2} + \frac{P_j}{(\epsilon_j \rho_j)^2} + \Pi_{ij} + R_{ij} \right) \nabla W_{ij} + F_{ext}/m_i \quad (78)$$

$$\Pi_{ij} = m_j \frac{(\mu_i + \mu_j) r_{ij}}{\rho_i \rho_j (r_{ij}^2 + 0.01h^2)} v_{ij} \quad (79)$$

$$R_{ij} = \frac{v_{max}^2}{c_s^2} \left| \frac{P_i}{(\epsilon_i \rho_i)^2} + \frac{P_j}{(\epsilon_j \rho_j)^2} \right| \left(\frac{W_{ij}}{W_{(\Delta P)}} \right)^4 \quad (79)$$

where $F_{ext} = \sum(F_f^{fs} + F_f^{fp})$ is the external forces including fluid-solid particle interaction force and fluid-structure interaction force, Π_{ij} is the non-artificial viscosity term with separate physical viscosity of each particle element derived in [130], $0.01h^2$ in the denominator is meant to avoid singularity, R_{ij} is the anti-clump term introduced into the momentum equation to prevent particle elements from forming into small clumps due to unwanted attraction [45], the maximum velocity of the fluid medium is given as $v_{max} = \frac{1}{10} c_s$, and ΔP is the initial particle spacing.

The fluid pressure is calculated under the assumption of weakly compressible flow [132]:

$$P = B \left(\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right) \quad (80)$$

where γ is a constant taken to be 7 in most circumstances, ρ_0 is the reference density and B is the pressure constant. The subtraction of 1 on the right-hand side of Eq.(80) is to remove the boundary effect for free surface flow [108].

For the fluid-solid particle interaction, the drag force acting on a solid particle (*i.e.*, a single DEM particle element) returned to a fluid particle element in SPH is determined as a partition of the drag force in proportion to the weight of each fluid particle element:

$$F_f^{fp} = -\frac{m_f}{\rho_f} \sum \frac{1}{S_i} F_p^b W_{fp} \quad (81)$$

$$S_i = \sum \frac{m_j}{\rho_j} W_{ij} \quad (82)$$

where superscript fp represents the interaction between fluid and particle and b is the buoyancy force.

For fluid-fluid interaction, the hydrodynamic force is directly computed through pure SPH algorithm. In this study, fluid-structure interaction is governed by Newton's Third Law in which the forces on the structure from the fluid and the forces on the fluid from the structure are equal in magnitude but opposite in direction. The interaction forces between fluid SPH particle elements and structure DEM particle elements evolve with the SPH algorithm. The density and the pressure for structure DEM particle elements remain unchanged at all times, and only their velocity and position evolve with time.

4.3 Interaction forces for structure in DEM

As the structure is made up of bonded solid particles to represent the inherent property of material, the bond forces between particles are dominant to be transferred in the structure phase. When the bond stress exceeds the threshold value of bond strength, the bond breaks and then the unbonded solid particle is not governed by linear parallel bond model any more. For structure-solid particle interaction, the contact force is the same as the one in solid particle-solid particle interaction, which is due to the same nature of DEM particles. In a similar way, the DEM particle elements in structure phase is evolved with SPH particle elements in fluid phase to

compute the interaction forces between fluid SPH particle elements and structure DEM particle elements

4.4 Flowchart of computational algorithms in SPH-DEM

The overall algorithm process is depicted in Fig.9. First of all, particle elements and boundaries are generated under initial conditions. Once the simulation begins, each particle element searches its surrounding particle elements through the linked-list scheme and interaction forces are computed. For structure particle elements, they are subjected to hydrodynamic forces from fluid particle elements, direct contact forces from solid particle elements and inherent bond forces from themselves. The bond forces determine the breakage of the bond if the excess of tensile strength is reached. The fluid particle elements are not only subjected to hydrodynamic forces but also under the reaction forces (e.g. drag forces and buoyancy forces) from solid particle elements using the technique of Shepard filter. In addition, to drag forces and buoyancy forces from fluid particle elements, direct contact forces also exist among solid particle elements. In terms of boundary treatment, boundary particle elements are specific for SPH particle elements through SPH algorithm. On the other hand, boundary lines work for DEM particle elements according to the linear contact model when DEM particle elements approaching to boundaries. After the calculations of interaction forces acting on each particle elements, its position, velocity and density are updated at each time step until the end of calculation.

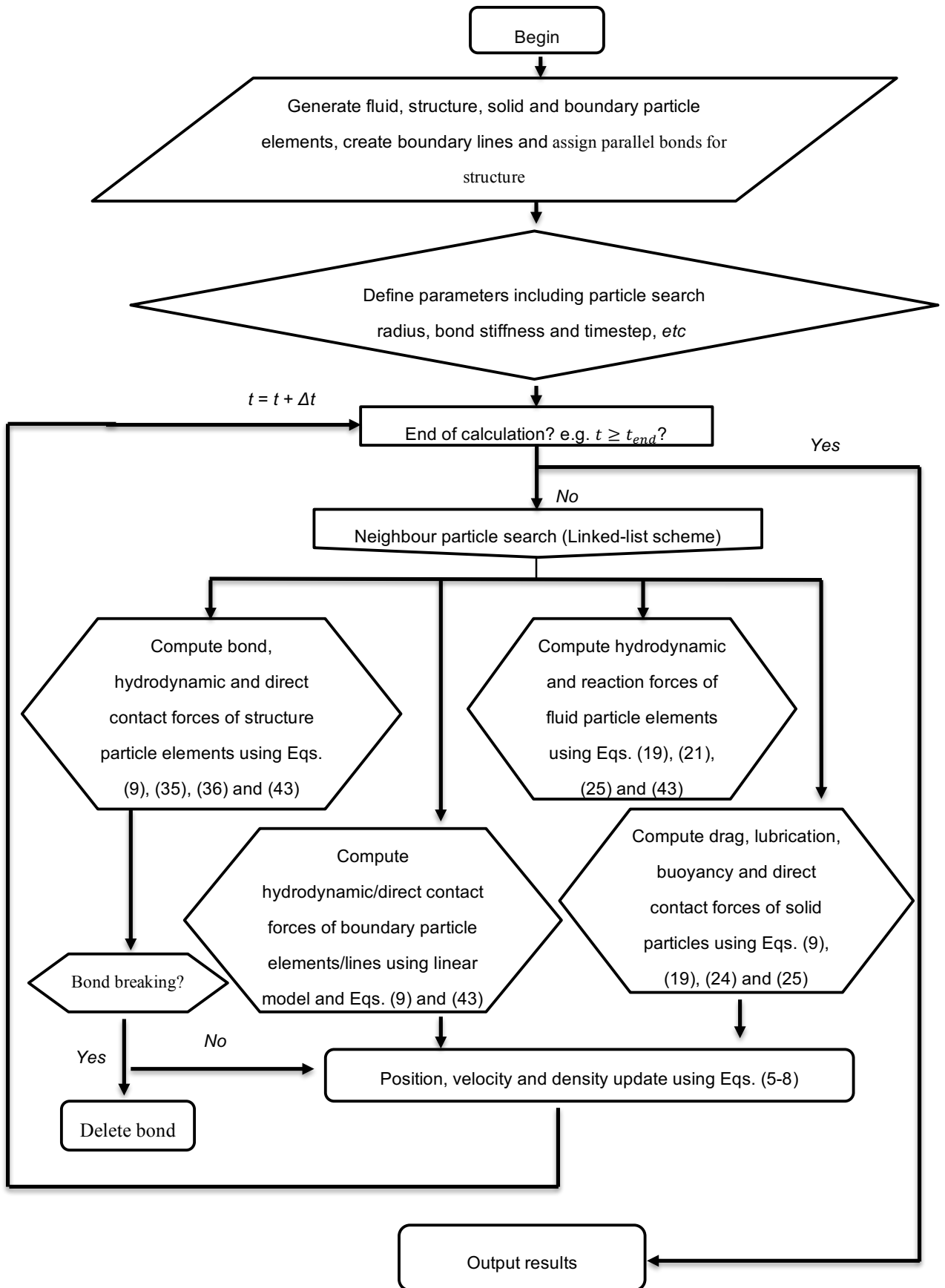


Fig. 9 Computational flow chart of the integrated particle model

5 Validations of SPH

5.1 Static tank test

A simple static tank test with an initial cubic packing of particles is set up in a 30mm×30mm tank, as shown in Fig.10. Two different kernel functions are investigated, *i.e.* cubic spline kernel and Wendland kernel. Under only gravity the particle distribution was observed for a time period of 1.0 second.

Each SPH particle is initialised with a hydrostatic pressure in accordance with the particle's position and its reference density. The density of each particle is then updated through equation of state in Eq.(44). The particle spacing Δp is 0.005m and the mass of particle in 2D simulation is described as:

$$m_i = \rho_i(\Delta p)^2 \quad (83)$$

where m_i is the mass of fluid particle.

The smoothing length is initially set as $h = 1.0 \times \Delta p$ for all cases and the maximum velocity of particle is assumed to be $v_{max} = \sqrt{2gD}$, where D is the depth of the fluid and g is the gravitational acceleration, $9.81m/s^2$. All the material and numerical properties are listed in Table.1

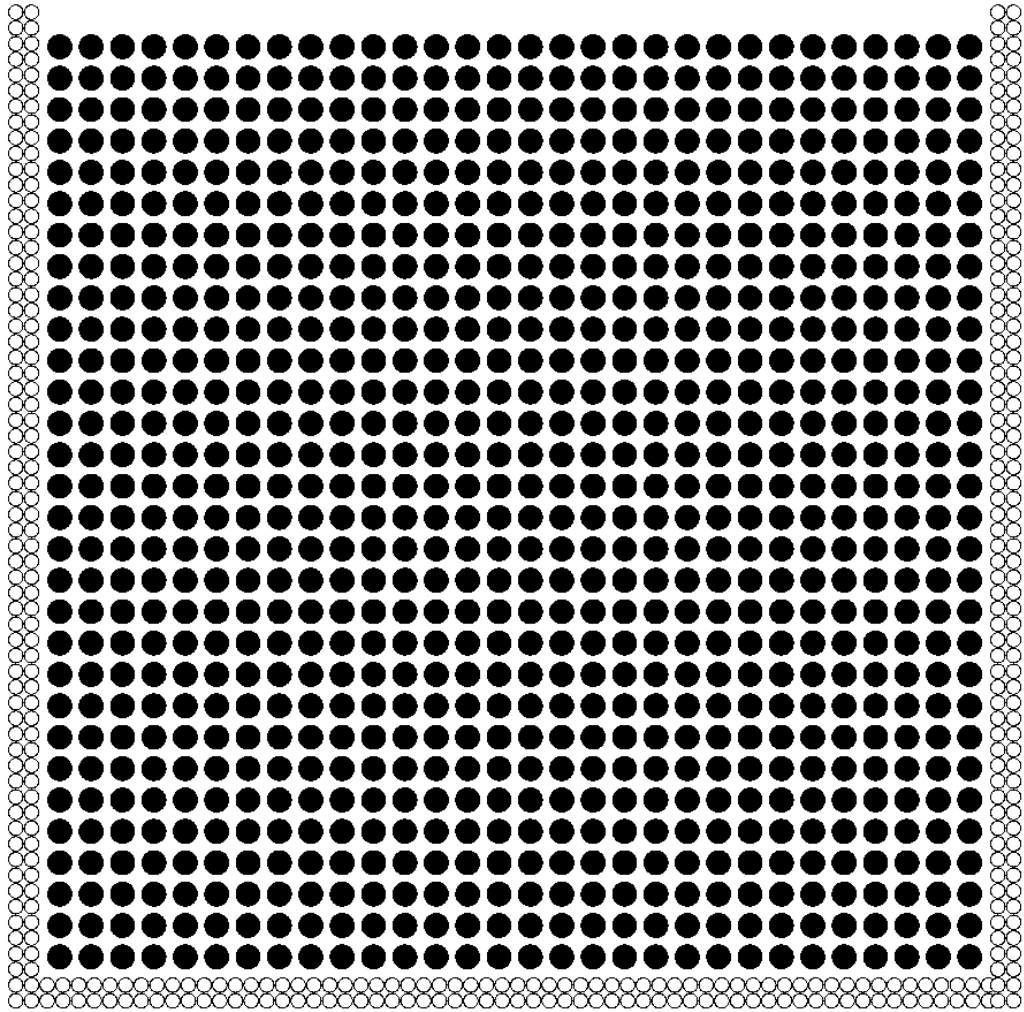
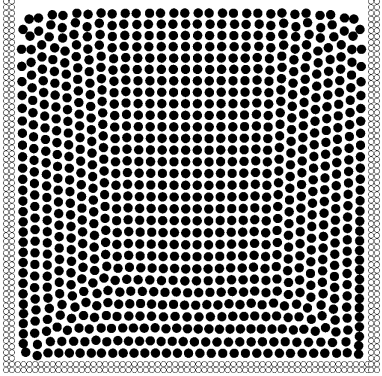
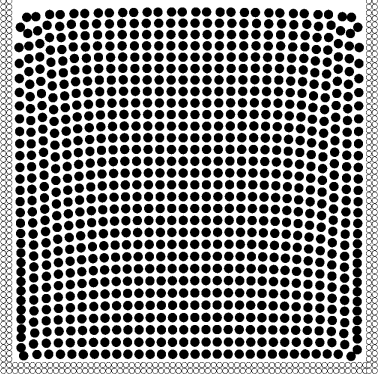
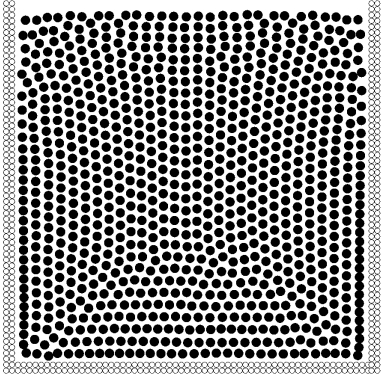
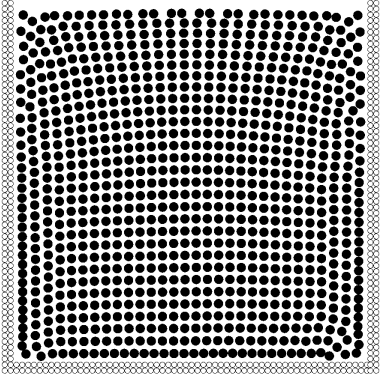
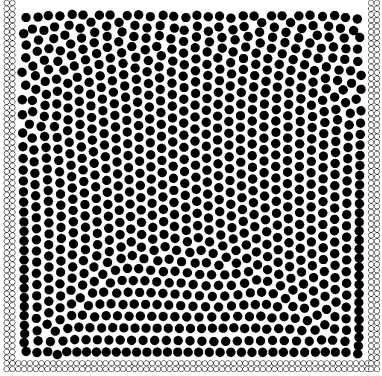
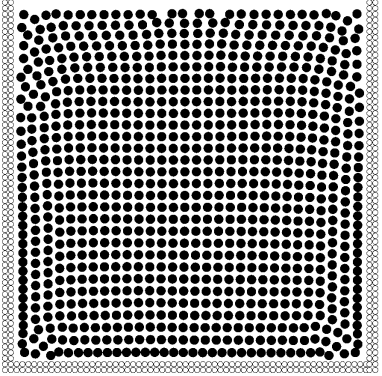
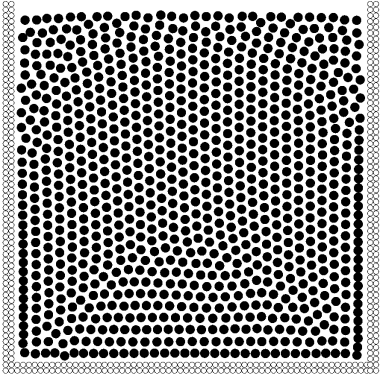
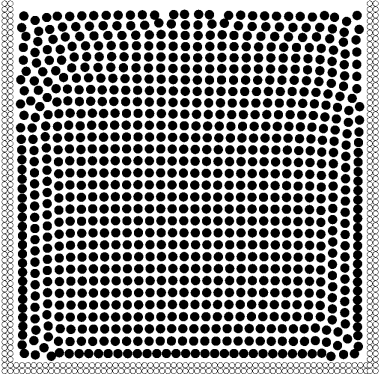


Fig. 10 Initial configuration of the static tank test
(SPH particles in black colour and boundary particles in white colour)

Table 1 SPH parameters used for the static tank test

Parameters	Values
Boundary particle spacing (m)	0.0025
SPH particle spacing (m)	0.005
Particle number	1278
Kernel function	Cubic spline/Wendland
Kernel smooth length (m)	0.005
Fluid density (kg/m ³)	1000
Fluid viscosity (Pa · s)	8.9×10^{-4}
Time step (s)	0.000004
Physical time (s)	1.0

Fig.11 shows the particle distribution at 0.2s time interval for cubic spline kernel and Wendland kernel, respectively. It can be seen that the particle distribution for the test using Wendland kernel nearly remains the same as the original particle distribution and only a small disorder is found at the corners of fluid, which is due to the boundary/interface deficiency. For the test using cubic spline kernel, the particles are packed orderly as well, but the particle distribution is not cubic any more, it and more likely becomes hexagonal after 0.2s. Even though the reason for this difference is not clear, both tests show good particle distribution without any particle cluster. Considering the Wendland kernel seems to produce better form of particle distribution, it will be used for all the simulations later.

Time	Cubic spline kernel	Wendland kernel
(a) t=0.2s	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.
(b) t=0.4s	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.
(c) t=0.6s	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.
(d) t=0.8s	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.	 A square grid of points with a central circular void. The points are arranged in a regular grid, and the void is a smooth, circular shape.

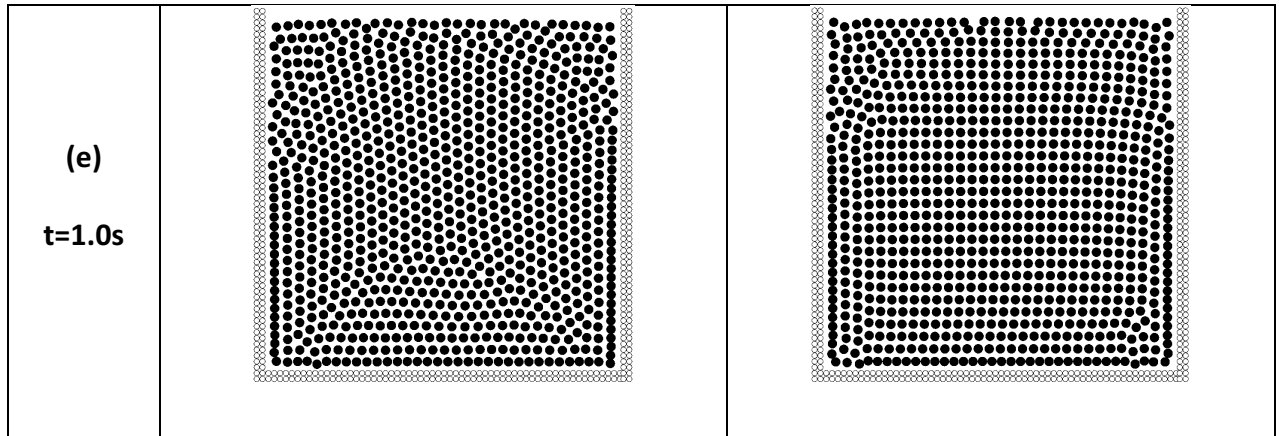


Fig. 11 Particle distribution during a period time of 1.0s using two different kernel functions.

5.2 Dam break test

5.2.1 Introduction

The case of a collapsing water column has been used in SPH studies [48, 142], therefore it is utilised here to validate the implemented SPH model. Besides the validation of SPH model, the effect of different smoothing length, $h = 1.0 \times \Delta p$, $h = 1.25 \times \Delta p$ and $h = 1.5 \times \Delta p$, is also examined. The geometry of the case is depicted in Fig.12 and the simulation parameters are listed in Table 2. The water column is initially adjacent to the left wall and is supported by a wall that is instantaneously removed when the experimental test starts. The water is thereby released into a dry channel. The SPH particles are initialised with hydrostatic pressure in accordance with the position and the density of each SPH particle derived by reversing Eq. (80). The distribution of density for the SPH particles is displayed in Fig. 13.

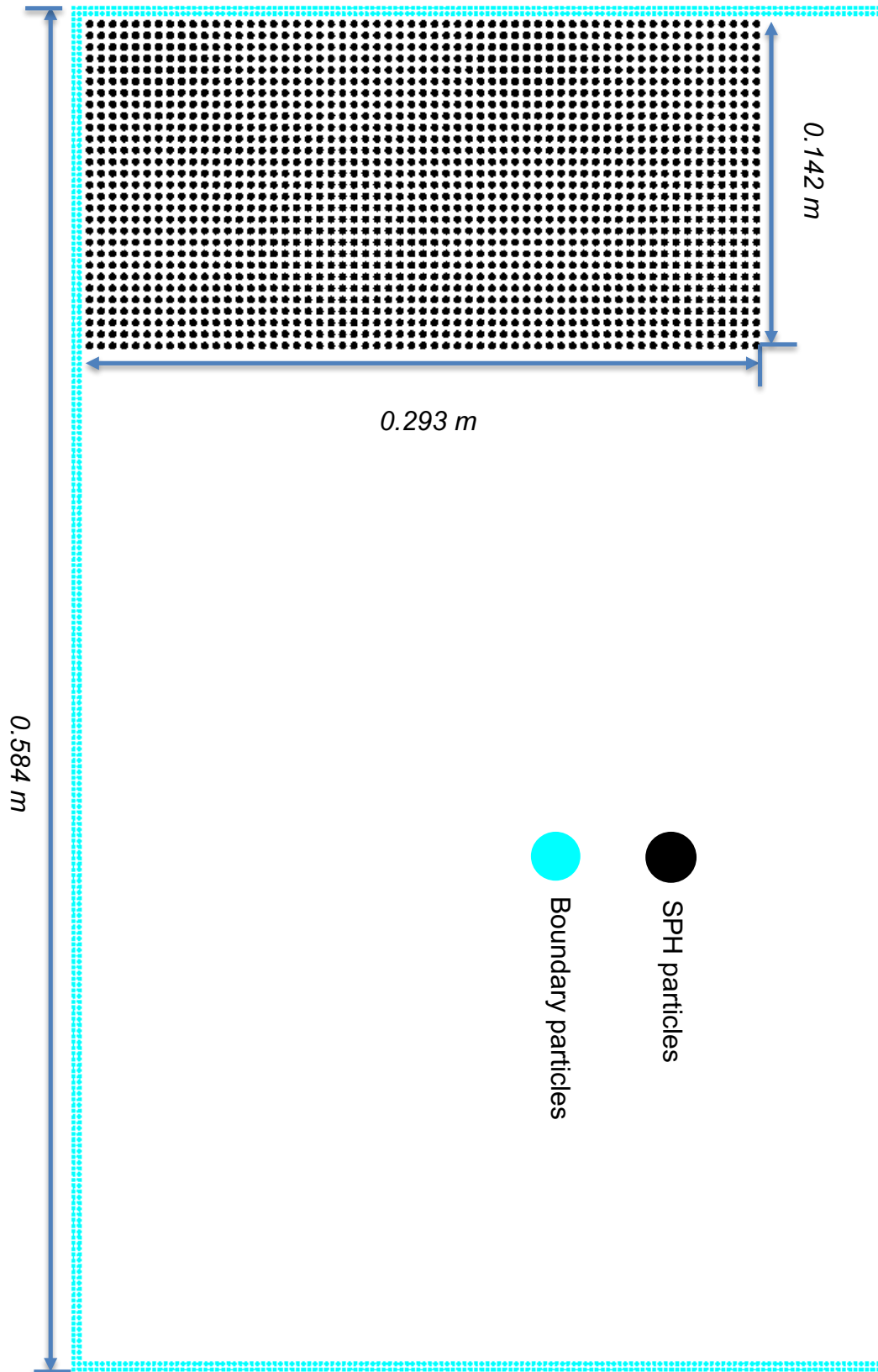


Fig. 12 2D SPH representation of the dam-break test

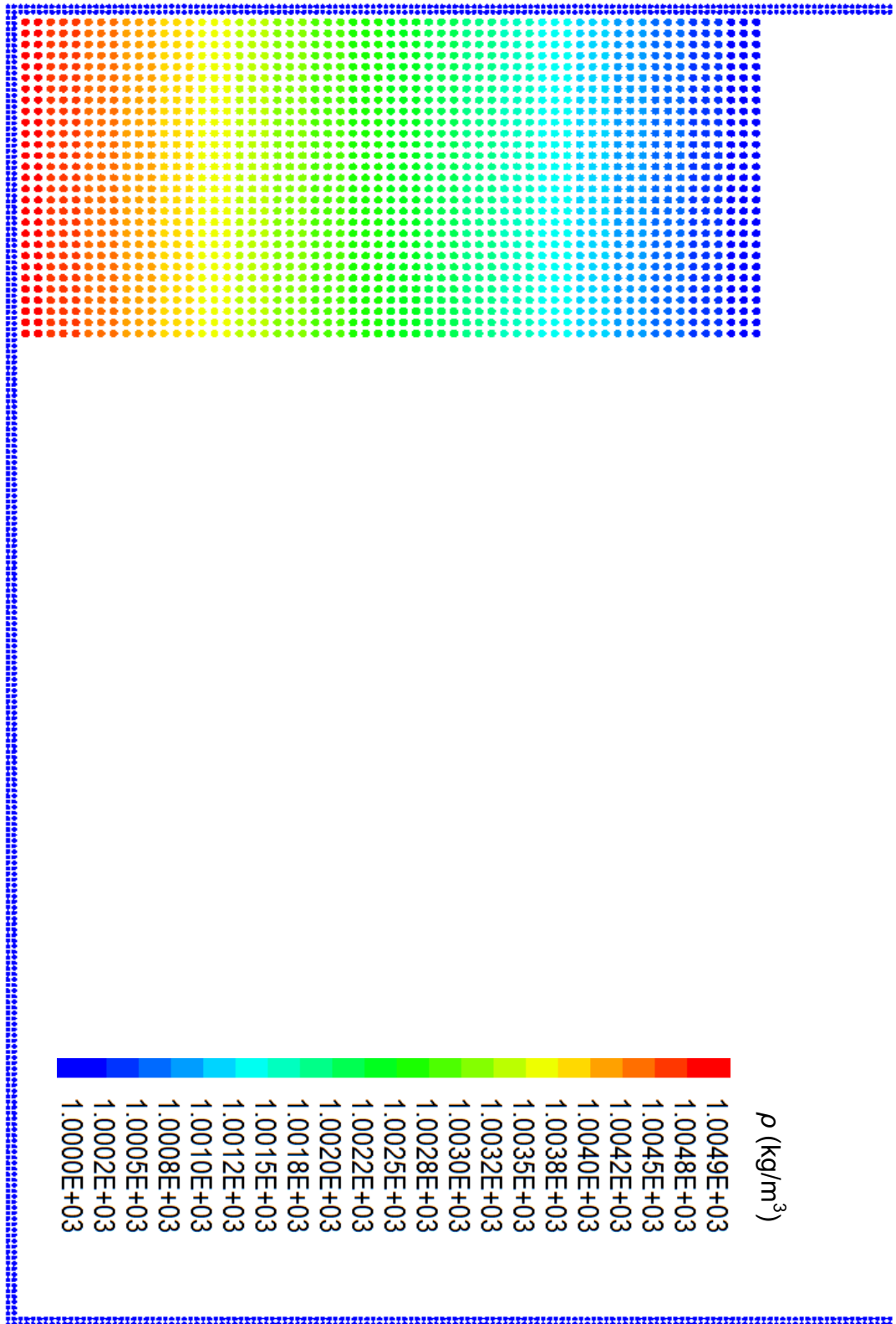


Fig. 13 Initial density of SPH particles with an assumption of artificial compressibility

Table 2 SPH parameters for the dam-break test

Parameters	Values
Boundary particle spacing (m)	0.0025
SPH particle spacing (m)	0.005
Particle number	2743
Kernel function	Wendland
Kernel smooth length (m)	0.005/0.00625/0.0075
Fluid density (kg/m ³)	1000
Fluid viscosity (Pa · s)	8.9×10^{-4}
Time step (s)	0.000004
Physical time (s)	1.0

5.2.2 Effect of different smoothing lengths

In Fig. 14, the SPH simulation results with a smoothing length of $h = 1.25 \times \Delta p$ are compared with experimental images as well as the numerical simulations using the moving particle semi-implicit method (MPS) for a time period of 1.0s with a time interval of 0.2s [48]. The collapsing water runs along with bottom wall with an increasing velocity at the leading edge at 0.2s (see Fig. 13a), and the accelerated water is then blocked by the right vertical wall thereby moving upwards at 0.4s (see Fig. 13b). At 0.6s, the SPH particles tend to reach the highest position with losing momentum energy which is offset by gravitation acceleration and then these SPH particles fall down to hit other SPH particles which still move along with bottom wall. At

1.0s, the movement of reflected SPH particles is gradually restricted by the left vertical wall. In general all the simulated flow patterns of water in SPH agree well with experiment and MPS.

Fig.15 shows the numerical results with different smoothing length. The flow patterns are almost identical before 0.6s. After 0.6s, those fluid particles repelled back by right vertical wall are gradually mixing with incoming fluid particles that are approaching right vertical wall. Due to this expected phenomenon, the simulations with longer smoothing length can search more surrounding particles to more accurately represent the fluid profile. It is apparent that results with $h = 1.25 \times \Delta p$ and $h = 1.5 \times \Delta p$ showed a good match in fluid profile from the beginning to 1.0s. In addition to the fluid flow profile, computational cost is another determining factor. More computational cost is required for the simulation with $h = 1.5 \times \Delta p$, and it produces better results than $h = 1.0 \times \Delta p$ (particularly at time 1.0s, see Fig.15e) but similar with $h = 1.25 \times \Delta p$. Therefore, smoothing length $h = 1.25 \times \Delta p$ is a better choice for numerical accuracy and computational efficiency and it is chosen for the rest of numerical simulations.

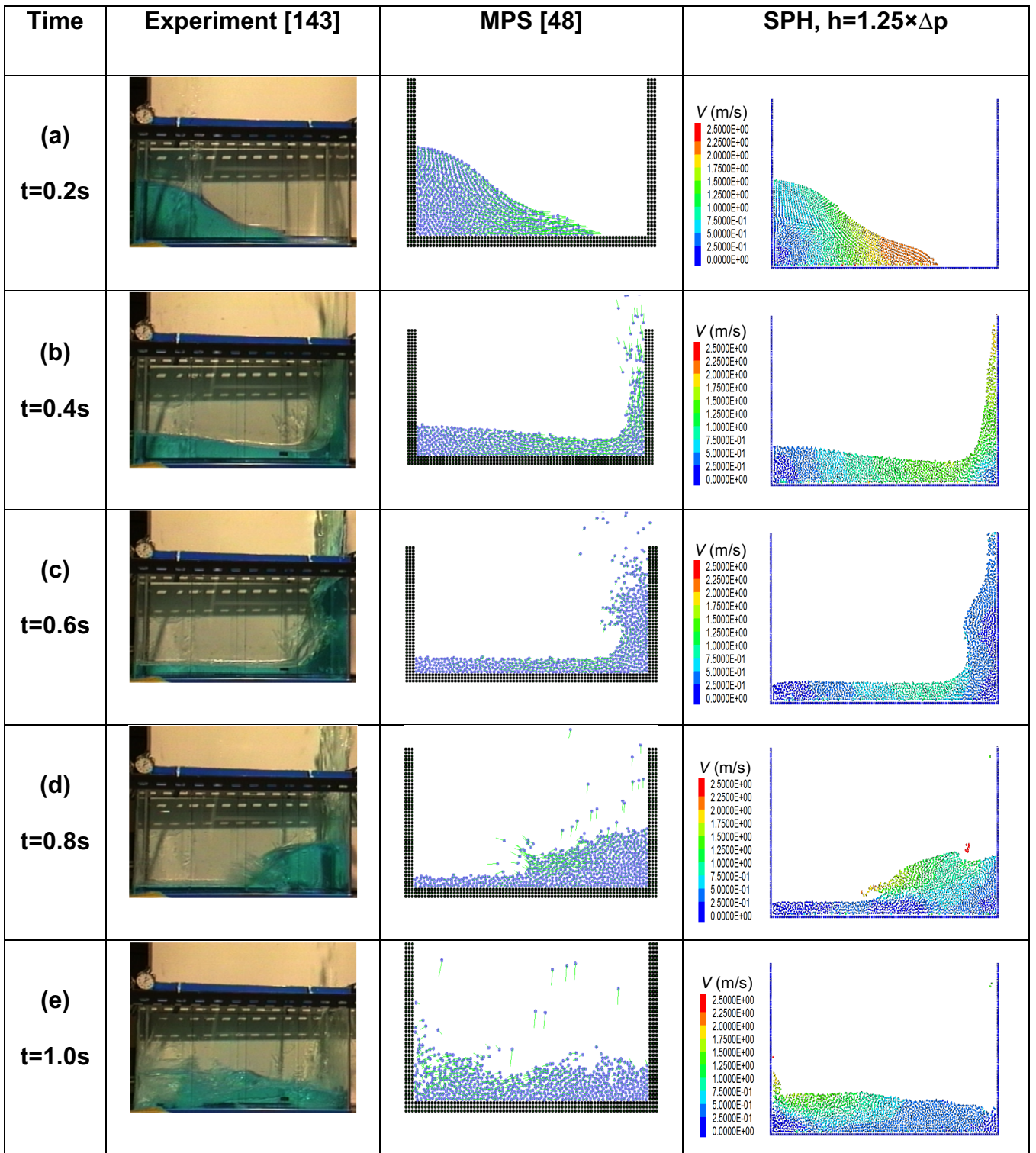


Fig. 14 Results from experiment [51], MPS [53] and SPH with $h=1.25 \times \Delta p$ for a time period of $t=1.0s$.

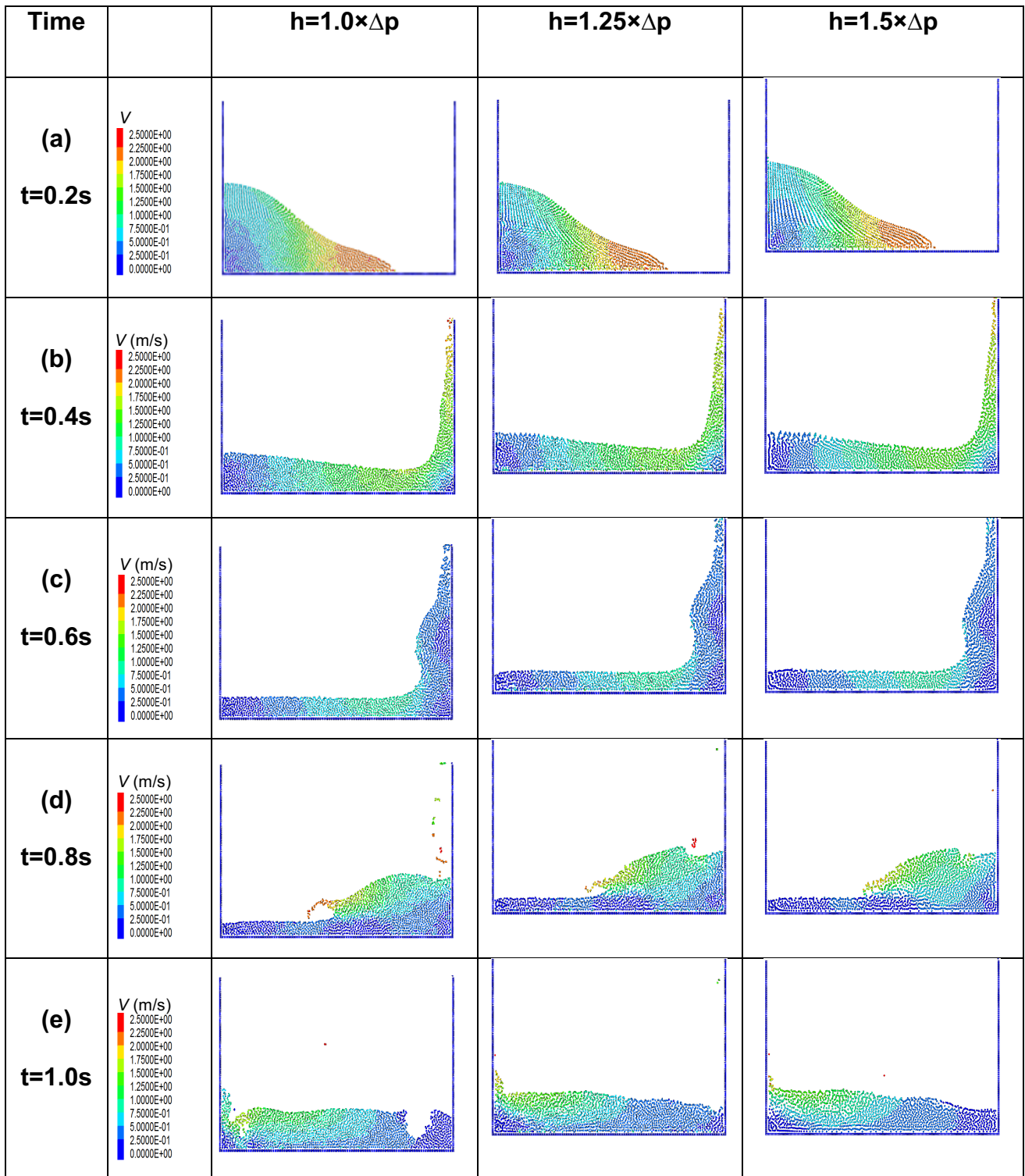


Fig. 15 SPH simulations with three different smoothing length for a time period of $t=1.0s$.

5.2.4 Effect of different particle resolutions

In this section the dam break test is simulated again using the same kernel function (Wendland kernel) and smoothing length ($h = 1.25 \times \Delta p$) but three different particle resolutions, *i.e.* particle spacing Δp . The particle spacing of 0.005m used before is chosen as a sample data, and two more different particle spacing, one is finer whilst the other one is coarser, are investigated for comparisons. The data for three particle resolutions are presented in Table.3, and the simulation results for physical time 1.0s are shown in Fig.16.

Table 3 Particle resolutions in the dam-break test

Parameters	Coarse	Medium	Fine
Boundary particle spacing (m)	0.003	0.0025	0.002
SPH particle spacing (m)	0.006	0.005	0.004
Particle number	2087	2743	4098

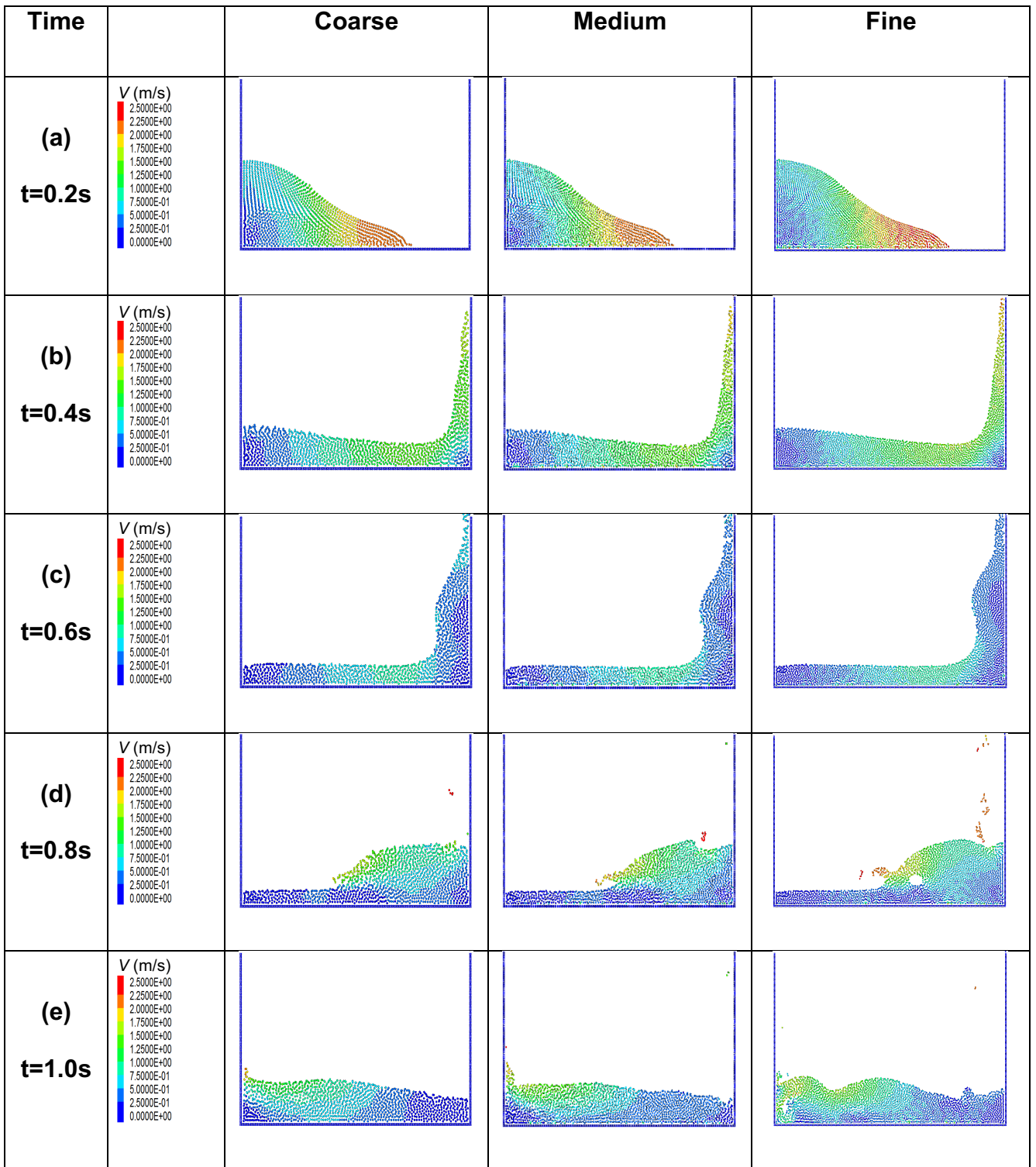


Fig. 16 SPH simulations with three different particle resolutions for a time period of t=1.0s.

It is evident in Fig.16 that the results for particle spacing of $\Delta p = 0.004$ show the best fluid flow profile and even capture the void at time $t = 0.8s$ and the curved wave at time $t = 1.0s$. Usually finer particle resolutions would give better results with more fluid flow profile details but at a cost of computational time. Therefore it is essential to balance the numerical accuracy and the computational cost, which highly depends on the kind of results that are expected to achieve. In this study, Wendland kernel and smoothing length in $h = 1.25 \times \Delta p$ are determined to be applied in the following simulation of fluid-structure interaction, and the particle resolution will be adjusted in accordance with the testing problem at a reasonable computational cost.

6 Validations of DEM

6.1 Dry dam break test

Solid particle-solid particle interaction in particulate flow is fully accounted for by DEM in this integrated particle model. Validation is carried out using the dry dam break test and the results are compared with previous modelling [111, 144] and experiments [144]. In the experiment from [144], solid cylinders with a diameter of 1 cm and a length of 9.9 cm are initially stacked in 6 layers with a hexagonal distribution. The cylinders are made of aluminium with a density of 2700 kg/m^3 , a Poisson's ratio 0.3 and a Young's Modulus of 69 GPa. The dimension of the tank is 26 cm in length, 10 cm in width and 26 cm in height. A plate is placed on the right-hand side of the stacked cylindrical columns and is quickly moved upward to trigger the movement of the cylinders under gravitational acceleration. A high-speed camera is used to record the transient behaviour of solid cylinders. A numerical model is constructed according to the initial configuration of dry dam break with a stack of solid cylinders, as shown in Fig.17. The friction coefficient of aluminium is set as 0.45, time step is 0.000001 and total simulated time is 0.5 s.

Fig.17 shows the obtained numerical results which are compared with previous experimental and DEM results available in the literature. The present numerical results seem to accurately capture the positions of the cylinders throughout the collapse process. It can be concluded that the

present unified particle model is capable of simulating the particle-particle interaction with a high accuracy

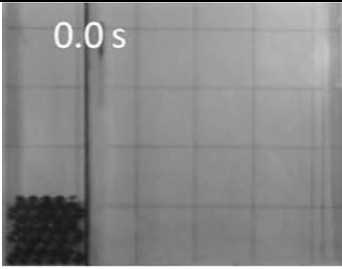
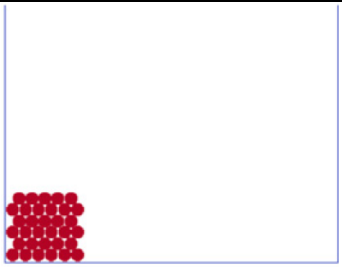
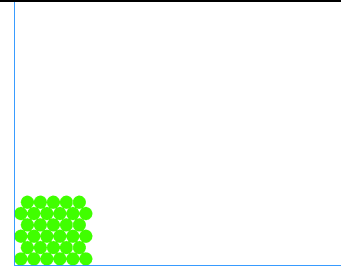
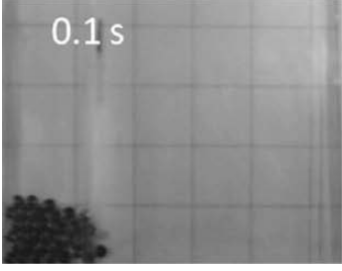
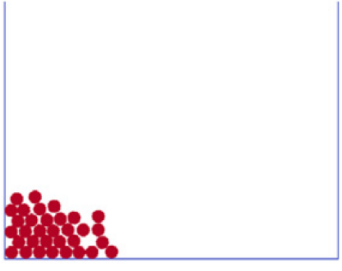
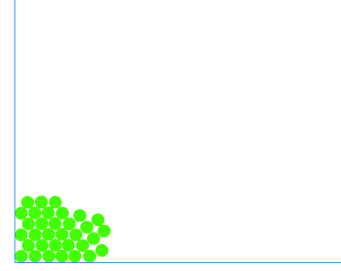
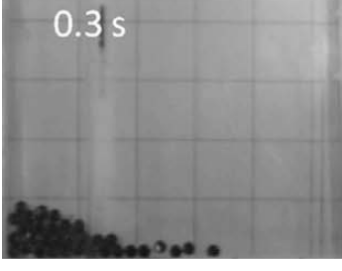
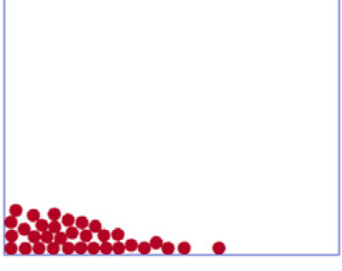
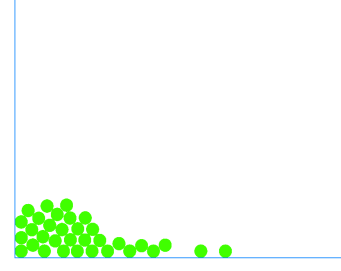

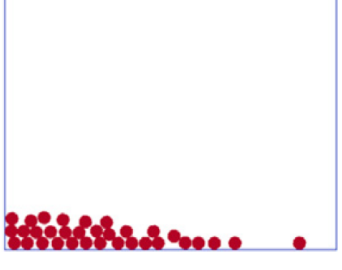
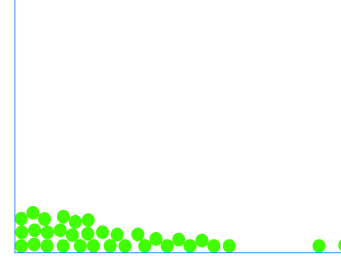
Time	Experiment [144]	DEM [111]	DEM (Present)
t=0.0s			
t=0.1s			
t=0.3s			
t=0.5s			

Fig. 17 Dry dam break test for a time period 0.5 s.

6.2 Tip-loaded cantilever beam test

To verify the capability of DEM in modelling the structure part in later FSI simulations, a tip-loaded cantilever beam test is studied in this section. Comparisons of deflections, stress distributions and final failure load are made to carefully evaluate the accuracy of the DEM approach in modelling structural deformation and fracture.

The material properties of cantilever beam are shown in Table 5 and the configuration of the beam is shown in Fig.18. The left end of cantilever is clamped and the other side of cantilever is under an increasing upward force F to give rise to a deflection.

Table 4 The list of material and particle properties

Material properties	Values
Density, ρ (kg/m ³)	2800
Ultimate tensile strength, σ_{ult} (MPa)	310×10^6
Young's modulus, E (N/m ²)	70×10^9
Poisson's ratio, ν	0.33
Particle radius, R (m)	0.0005
Bond radius, \bar{R} (m)	0.0005
Cantilever length, L (m)	0.201
Cantilever height, h (m)	0.006196

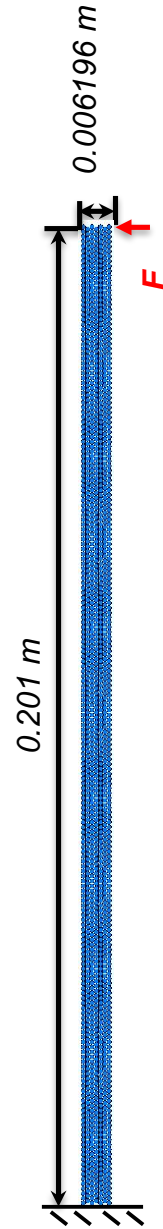


Fig. 18 Configuration of cantilever under single point load in DEM

The deformations of cantilever under three sets of upward forces, 50 N, 500 N and 5000 N, applied on the right bottom tip of cantilever are compared with analytical solutions [145] in Table 6. The deflection is measured when the model reaches an equilibrium state that the ratio of the unbalanced force (*i.e.* the sum of contact force, body force and applied force) to the sum of body force and applied force is extremely small, *e.g.* 1×10^{-7} . The results from DEM model and analytical solution are almost fully matched with acceptable small errors which may be due to the fact the load is applied at the centre of the particle not the exact edge of the real beam.

Table 5 Deflections for the tip-loaded cantilever beam test

Load (N)	deflection in DEM model (m)	deflection in analytical solution (m)	Error
50	9.7533E-05	1.002417E-05	2.78%
500	9.7533E-04	1.00247E-04	2.78%
5000	9.7533E-03	1.002286E-03	2.76%

In addition, the same test of cantilever beam at load 5000N is carried using FEM software ABAQUS in order to compare the stress distribution. The element size in FEM is the same as the particle radius in DEM. It can be seen from Fig.19 that the distribution of stress component σ_{11} in FEM is nearly identical to the one in DEM. The maximum stresses in both methods are also very close with an error of 0.24%. This further confirms that the DEM model can accurately predict the structural deformation.

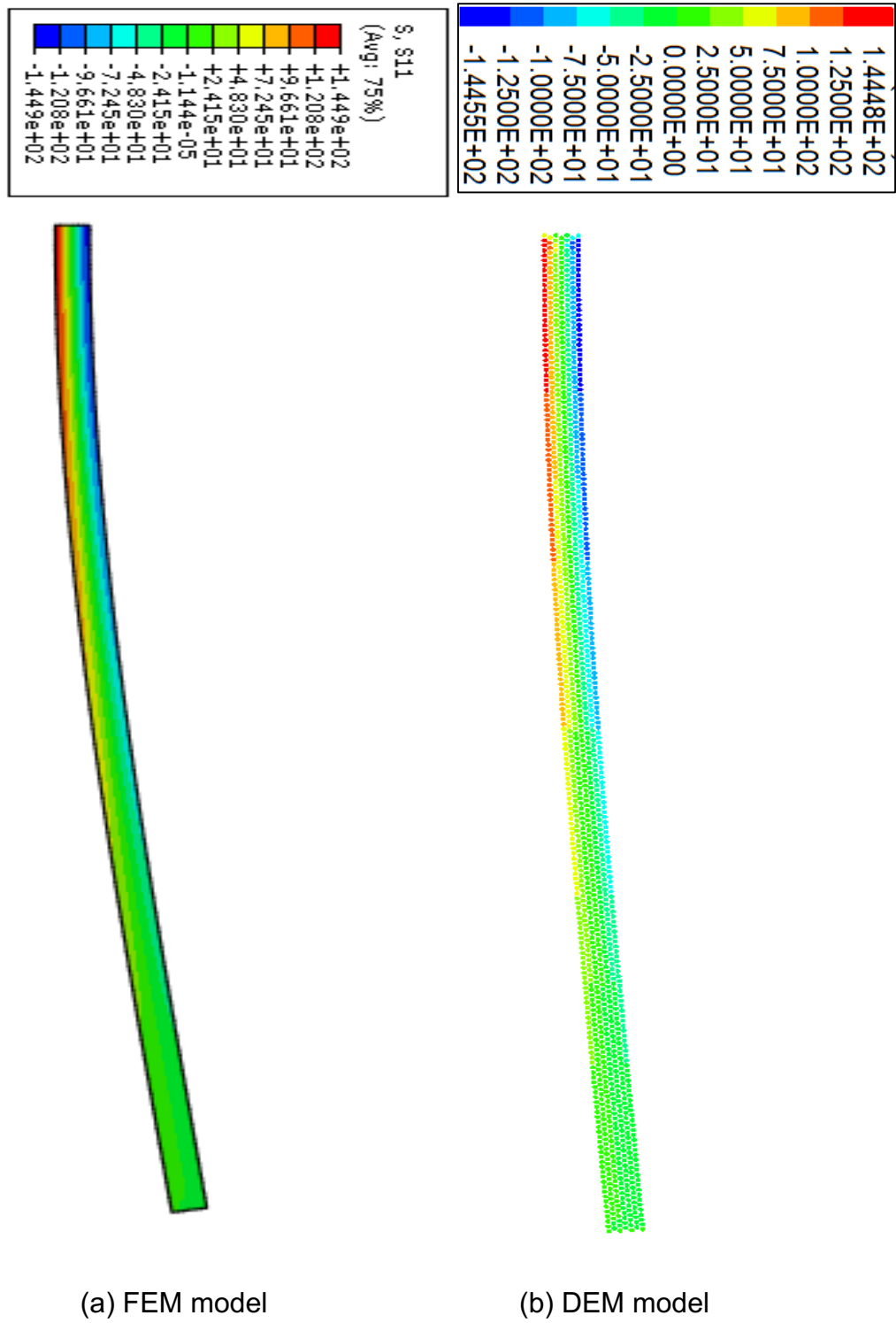


Fig. 19 Distribution of stress σ_{11} in cantilever beam at load 5000N

To test the failure of the beam, the incremental loading approach used above is replaced by assigning a constant and very small upward velocity $v = 0.02m/s$ to the particle at the right bottom end of the beam. This small loading velocity is chosen to ensure the structure under quasi-static loading condition till the final failure [134]. The simulation is stopped immediately once a bond breaking occurs. The obtained force at right bottom end of the beam is compared with analytical solution according to:

$$\sigma_{ult} = \frac{PL}{Z} \quad (84)$$

$$Z = \frac{bh^2}{6} \quad (85)$$

where Z is section modulus, b is the thickness of beam which is unit in 2D simulations. In this test, the cantilever beam is assumed to fail when maximum stress is equal to ultimate tensile strength. In Table. 6 the maximum applied load obtained from the DEM model shows good agreement with the applied load computed from Eqs. (84) and (85). It should be note that the DEM prediction is slightly higher the theoretical one which is calculated under the assumption that the beam is still perfectly straight at failure.

Table 6 Maximum applied load for the tip-loaded cantilever beam test

	Analytical	DEM	Error
Maximum applied load P (N)	9868.66	10322.1	4.595%

7 Validations of SPH-DEM

7.1 Single particle sedimentation test

Particle sedimentation has been extensively studied and verified [146, 147], and will be used to validate current integrated particle model for fluid-particle interaction. In this section, a case with a single particle settling in the fluid is simulated first and then the interaction between multiple particles and fluid is further investigated later. In this simulation, a particle with a density of 1250 kg/m^3 and a radius of 0.00125 m is initially placed in a box with a width of 0.02 m and height of 0.06 m as shown in Fig.20. The centroid of particle has a vertical distance of 0.04 m to the bottom of the box. The box is filled with fluid with a density of 1000 kg/m^3 and viscosity of $0.01 \text{ Pa}\cdot\text{s}$. The particle falls down due to the gravitational acceleration of 9.81 m/s^2 until it hits the bottom of the box. A total physical time of 1 second is simulated. For numerical parameters, the boundary particle spacing and fluid particle spacing are 0.00125m and 0.0015m , respectively. The boundary particle has smaller spacing in order to numerically avoid particle penetration. The Wendland kernel is applied with a smoothing length 0.003m and the time step is set to be 0.000002s .

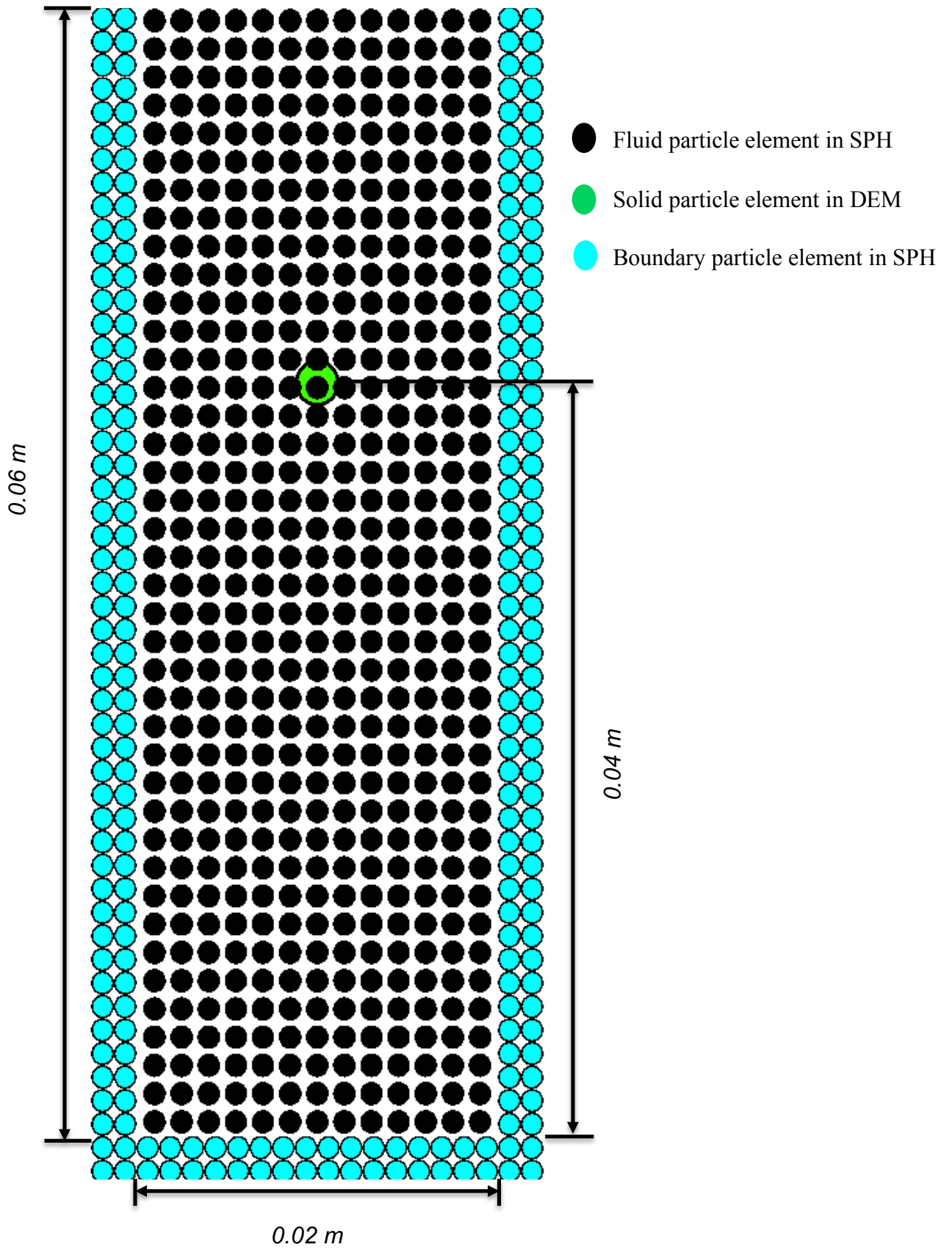
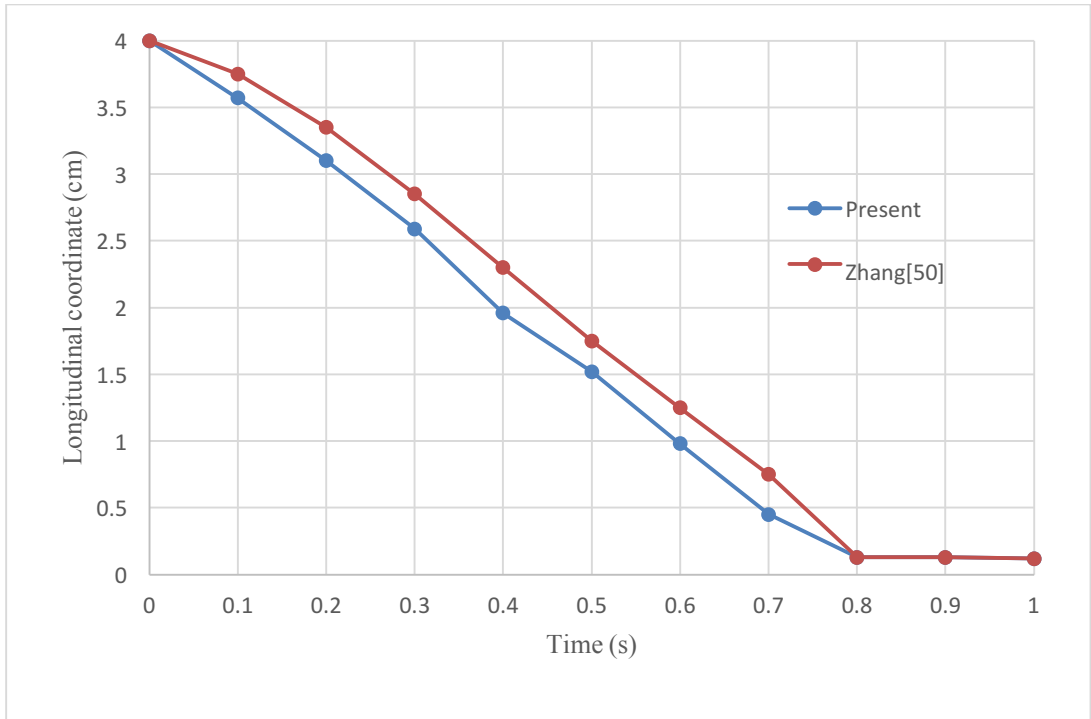
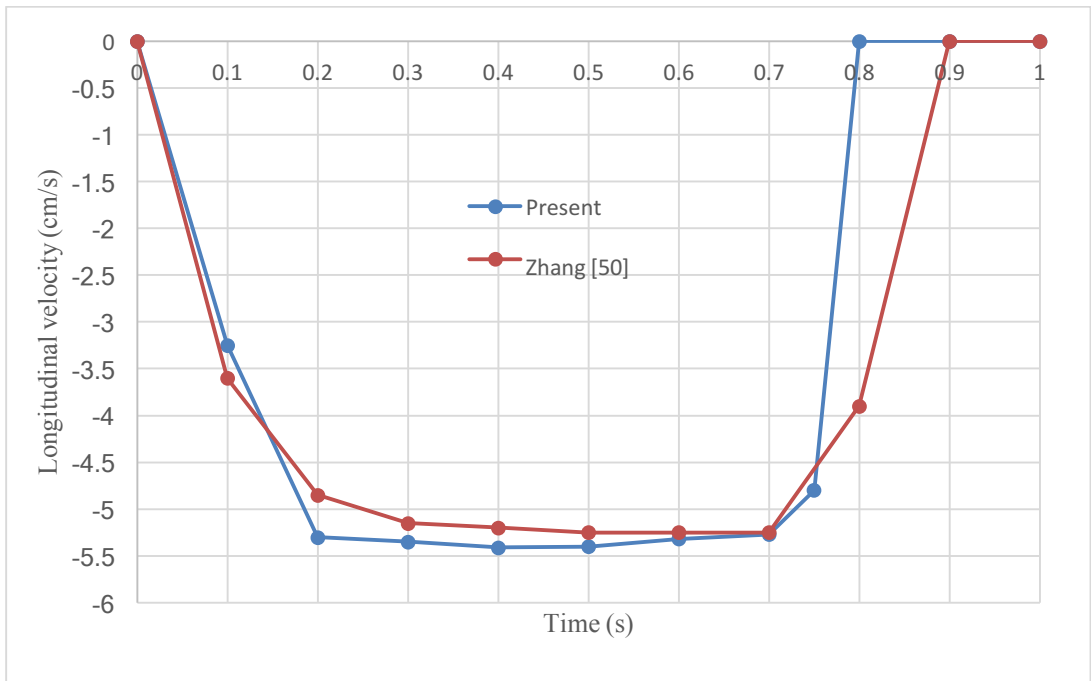


Fig. 20 Configuration of single particle sedimentation test

In Fig.21, longitudinal coordinate and longitudinal velocity of the particle are compared with numerical results from other researchers using immersed particle method (IBM) and Lattice-Boltzmann method (LBM) [147]. In general, the results obtained from the present SPH-DEM model almost match with those of IBM-LBM, and a minor difference is found at $t = 0.8s$ when the particle settles down to the bottom. This may be caused by the assumption of compressible flow used in current SPH method, and SPH particle elements can interact with each other with minor compression and expansion at different time, which can cause the fluctuation of particle element's velocity to affect the calculation of drag force. In addition, the restriction in the ratio of the resolution of fluid particle element to the diameter of solid particle element has been reported in [112] in terms of the fluid resolution length scale, which is one of the main assumptions in locally averaged Navier-Stokes (AVNS) equations. When a smoothing length is large enough, a smoother porosity field will be produced. On the other hand, a much finer fluid resolution with shorter smoothing length can result in less smoothness of porosity field. This confirms that the calculated porosity field is relatively larger, so that the solid particle element with faster terminal velocity drops downward.



(a)



(b)

Fig. 21 Longitudinal coordinate (a) and velocity (b) against time

7.2 Two phase dam-break test

A 2D simulation of two-phase dam-break test is carried out to further validate the proposed model. The initial configuration of the test is depicted in Fig.22. In this simulation, solid particles with a density of 2500kg/m^3 and an identical diameter of 0.0024m are randomly packed and aligned with the left and bottom boundaries of the reservoir and the moving boundary. The volume of the assembly of solid particle elements is estimated to be equivalent to 200g in total mass, same as in the experiment and 3D simulations in [138]. It should be noted that the mass of solid particle elements in 2D simulations is different from that in 3D simulations or experiments in [138]. In 2D simulation, the volume of solid particle is equivalent to the product of the area of solid particle and the unit thickness. Fluid particle elements with a density of 1000kg/m^3 and viscosity of $8.9 \times 10^{-4} \text{P} \cdot \text{s}$ are orderly distributed with a height of 0.1m and a width of 0.05m . The solid particles, each of which is represented by a DEM particle element, are completely immersed within the fluid. It should be noted that the overlap between solid DEM particle elements and fluid particle elements is due to the visualisation of SPH particle elements and has no effect on the simulation. When the solid DEM particle elements reach equilibrium after few cycles (*e.g.* no more energy dissipation), the simulation begins and the moving boundary moves upward at a constant velocity of 0.68m/s in the Y direction to initiate the movement of the mixture of solid particles and fluid in the X direction. The total physical time is 0.2s and the numerical timestep is set to be $2.0 \times 10^{-6}\text{s}$. The boundary particle spacing and fluid particle spacing are 0.0015m and 0.0024m , respectively, and the Wendland kernel is applied with a smoothing

length 0.003m. The behaviour of wave fronts is captured after quick removal of the dam and numerical results are compared with other experimental and numerical data from [138].

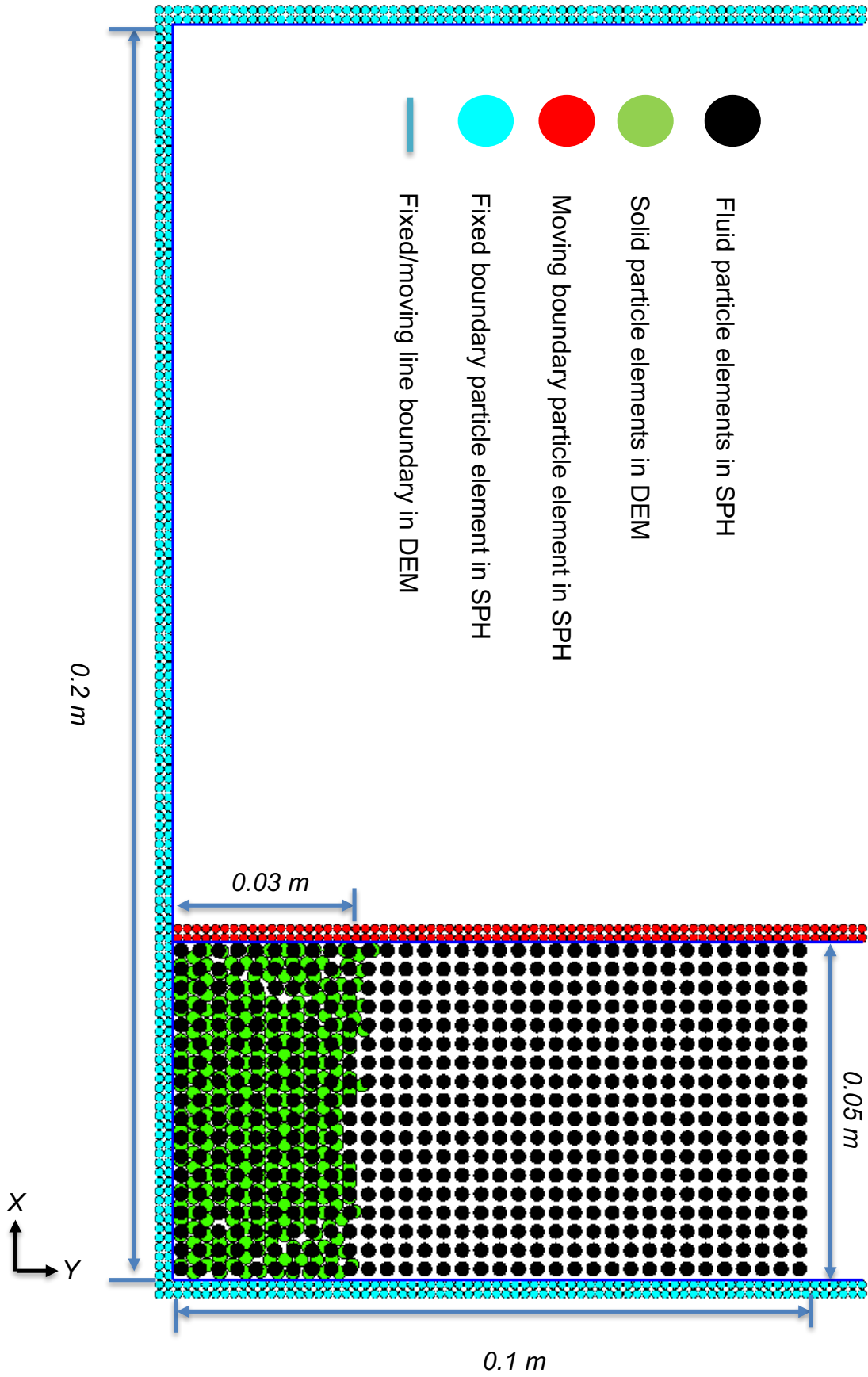


Fig. 22 2D representation of the two phase dam-break test

In this test, the dynamic behaviour of solid particles and fluid at the early stage of dam-break flow is observed and snapshotted at a time interval of 0.5 s. Fig.23 shows the numerical results in comparison with experimental and other researcher's numerical results. As the moving boundary starts moving upward, there is no restriction to inhibit the movement of fluid and solid particles. Subsequently fluid drags solid particles to move in the flow direction. Compared to sample experimental and numerical results, the flow pattern of either solid particles or fluid seem to match well at $t=0.05$ s, 0.10 s and 0.15 s. However, at time $t=0.2$ s, the solid particles and fluid move faster and the wavefront in the current study hits the boundary wall earlier. The present study is in 2D, so the forces acting on a solid particle from other solid particles as well as the fluid in the 3rd direction (*i.e.* thickness direction) is not counted, which subsequently should have caused differences in the movement of solid particles. In addition, in the experimental study [138], the diameters of solid particles are not constant, though the mean diameter of solid particles is 0.0027 m, which is slightly greater than the constant diameter used in the current study. Even though the constant diameter of solid particles can bring benefit in producing a smooth and stable porosity field, they may affect the overall interactions between solid particles.

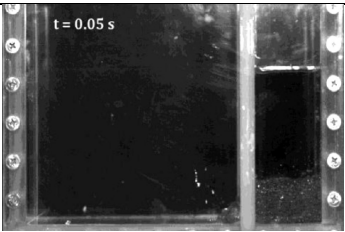
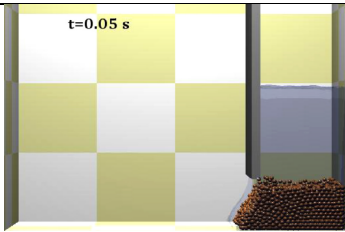
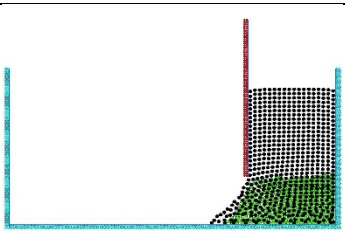
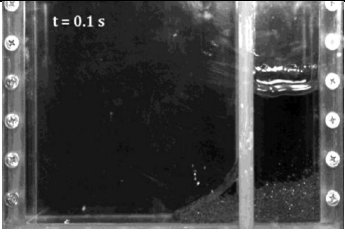
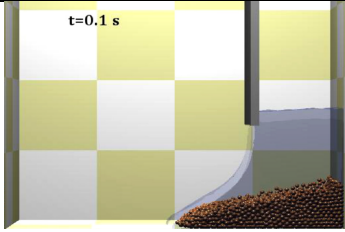
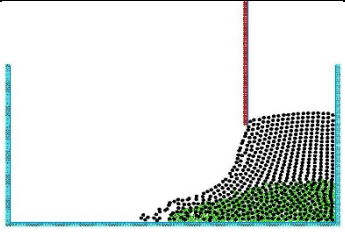
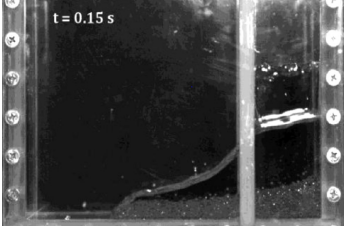
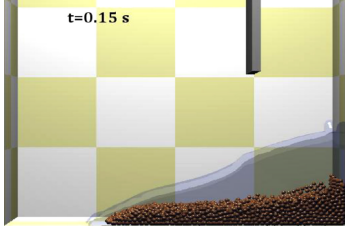
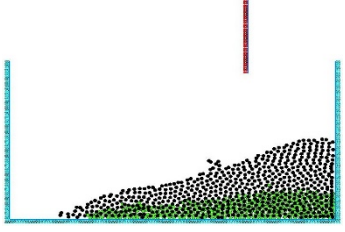

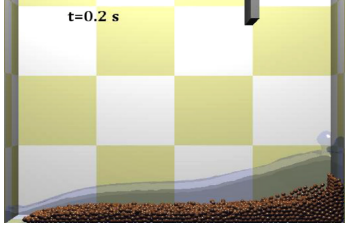
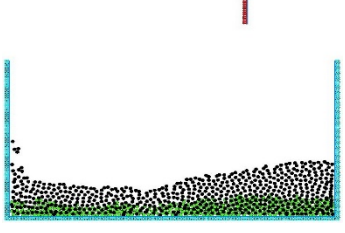
Time	Experiment [138]	SPH-DEM [138]	SPH-DEM (Present)
t=0.05 s			
t=0.10 s			
t=0.15 s			
t=0.20 s			

Fig. 23 Two phase dam-break test for a time period of t=0.2s

Next, two dimensionless numbers are introduced to make a quantified comparison for the propagation of wavefront:

$$z^* = \frac{z}{a} \quad (86)$$

where z is the position of wave front in x -direction, a is the width of dam, which is 0.05m

$$t^* = t\sqrt{2g/a} \quad (87)$$

where t is the physical time and g is the absolute value of gravitational acceleration. Fig.24 shows the normalised front wave position before touching the left end wall against the characteristic time. It is noted that the fluid in authors' simulation moves slightly quicker than that in experiment after the release of moving boundary, hence for better comparisons, the last data point in the author's results is taken at the time when the wavefront hits the left end wall. In the author's results, it's a difficult to judge an accurate position of the front wave as fluid particle elements in the area of front wave do not completely move in order after interacting with solid particles. Especially for time at 0.1s, a clearly visible void at front wave area can be seen. As a result, the accuracy of front wave position cannot be guaranteed, as it is sacrificed by assigning the most front fluid particle as the front wave position. In spite of this, the overall trend of the front wave positions is acceptably close to those from experiment and other numerical results.

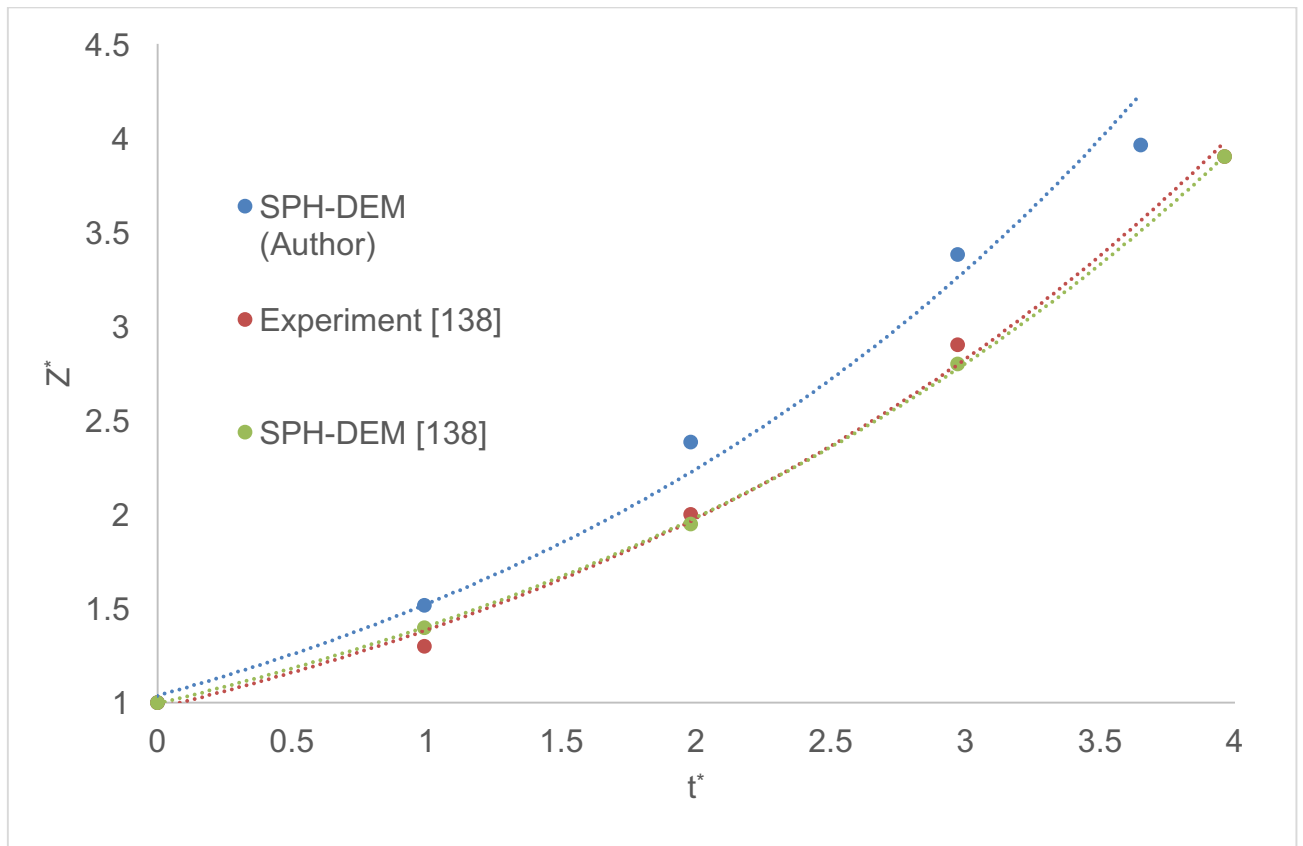


Fig. 24 The normalised front position against the characteristic time

7.3 Dam break with top-fixed elastic gate

The validation test for the coupled SPH-DEM model is to simulate the water flow in the elastic gate problem for comparison with experimental data and numerical results using a coupled SPH-SPH method [23]. The initial configuration is illustrated in Fig.25 and the simulation parameters for this validation case are listed in Table 7. The top end of elastic gate in purple is fixed and the other end is free to move. The bonded DEM particles representing the elastic gate are distributed in a hexagonal pattern and particle and bond stiffness is determined according to Eqs. (38-40) and (42). The SPH particles for water are initialised with hydrostatic pressure, and there is no pre-existing stress and deformation for bonded DEM particles.

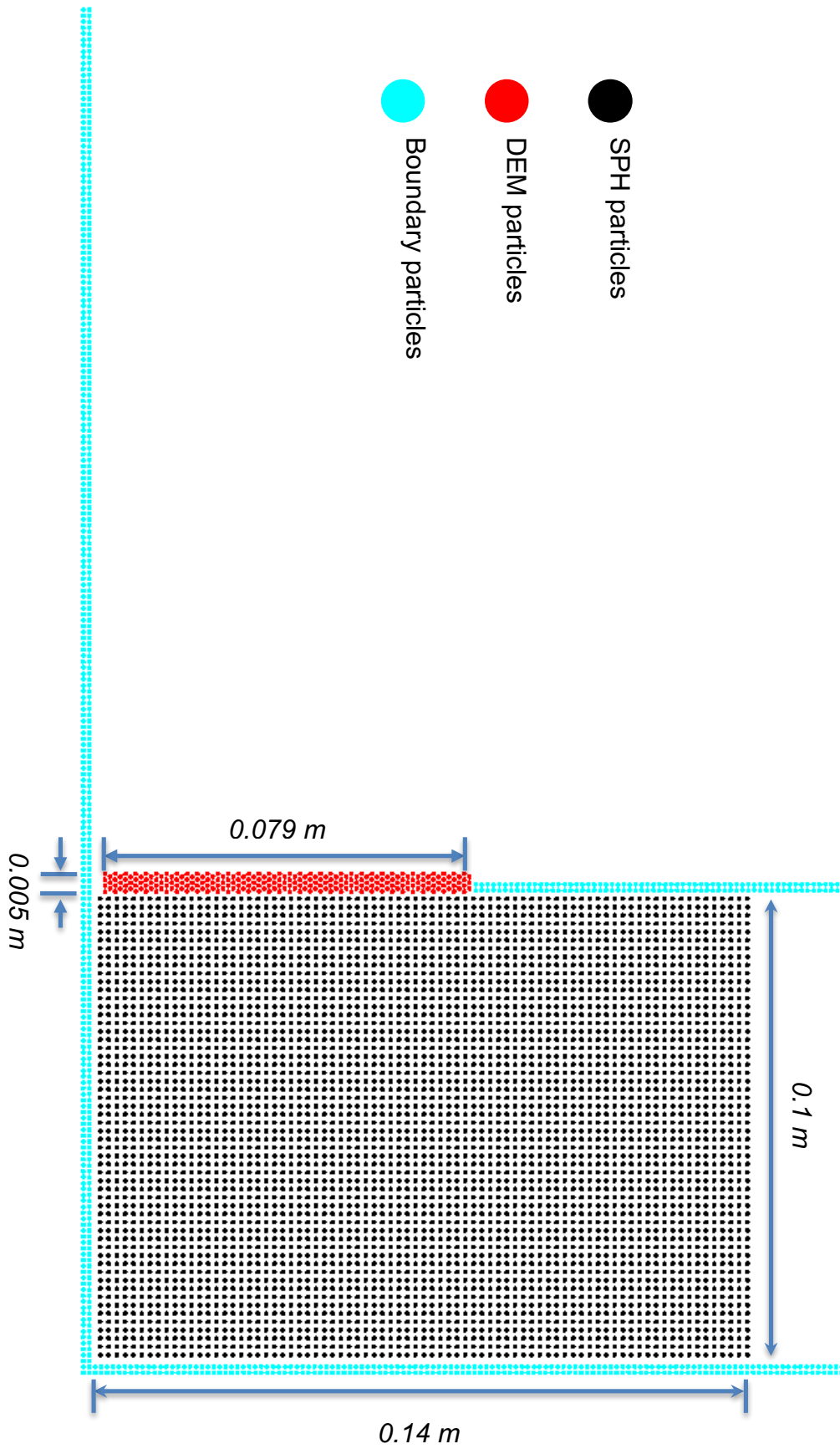


Fig. 25 Configuration of 2D elastic gate test in a coupled SPH-DEM model

A comparison of numerical results from the present SPH-DEM model against experimental data and numerical results from the coupled SPH-SPH model is shown in Fig.26. In the coupled SPH-SPH model, the solid dynamics is simulated through an incremental hypo-elastic relation. Compared to the experiment snapshots, the deformation of the elastic plate and the vertical displacement of free surface of water are generally well predicted by the coupled SPH-DEM model. The maximum deformation of the plate is at top end and behaves almost as a rigid body without deformation at bottom end of the plate. It should be noted that after removing the hammer, which fixes the elastic plate against water pressure immediately in the experiment, water leakage besides the elastic plate is observed. Owing to this leakage, the water pressure acting on elastic plate is lower than that in the SPH-DEM simulation and consequently the vertical displacement of the free surface in the SPH-DEM simulation is larger than the experimental results. In comparison with both experimental and SPH-SPH results, both larger deformation and higher vertical displacement of the free surface in SPH-SPH and SPH-DEM results make sense without water leakage. Due to the decreasing hydrodynamic pressure of water, the deformation of the plate tends to be smaller after 0.16s in the SPH-DEM model, and as a result the vertical displacement of free surface changes more slowly.

Table 7 Parameters for SPH-DEM modelling of the elastic gate test

Parameters	Values
Boundary particle spacing (m)	0.00125
SPH particle spacing (m)	0.00175
DEM particle size (m)	0.00125
Particle number	6648
Kernel function type	Wendland
Kernel smooth length (m)	0.0021875
Fluid density (kg/m ³)	1000
Fluid viscosity (Pa · s)	8.9×10^{-4}
Gate density (kg/m ³)	1100
Gate elastic modulus (MPa)	12.0
Gate Poisson's ratio	0.33
Time step (s)	0.000004
Physical time (s)	0.4

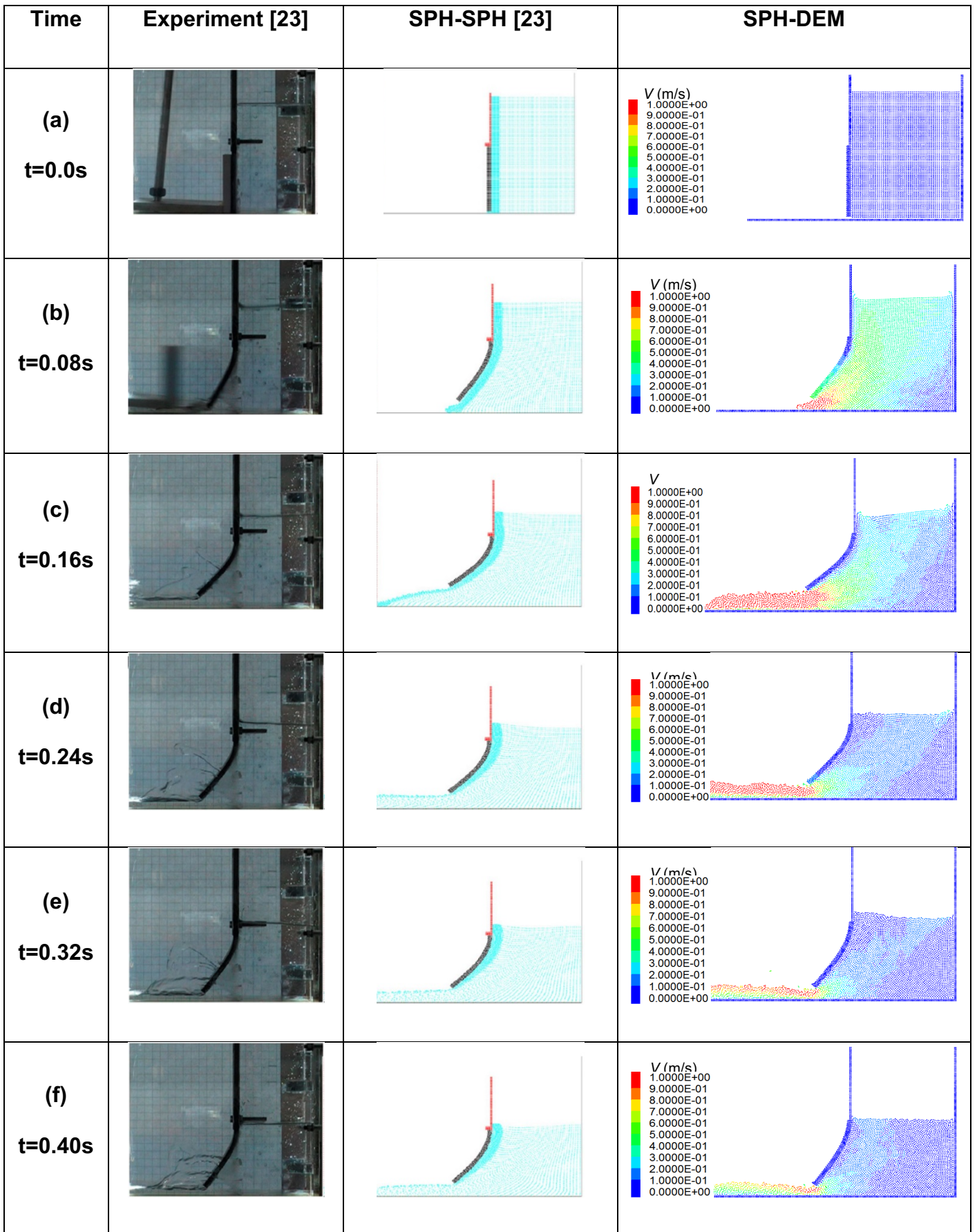
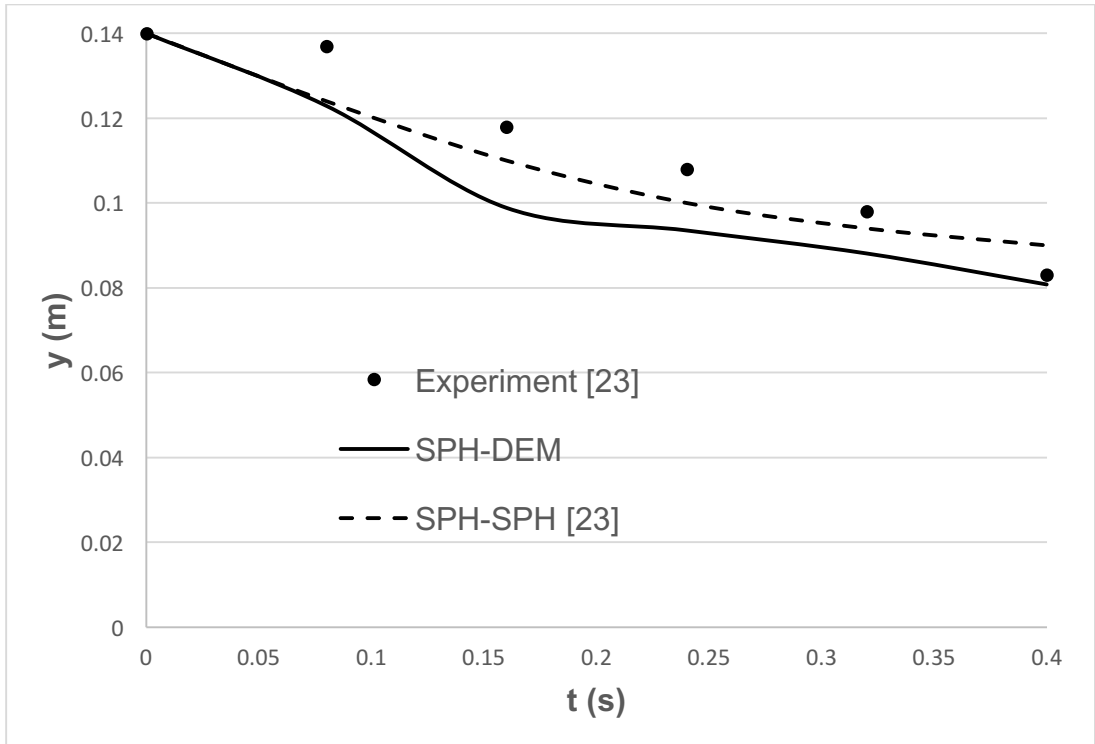
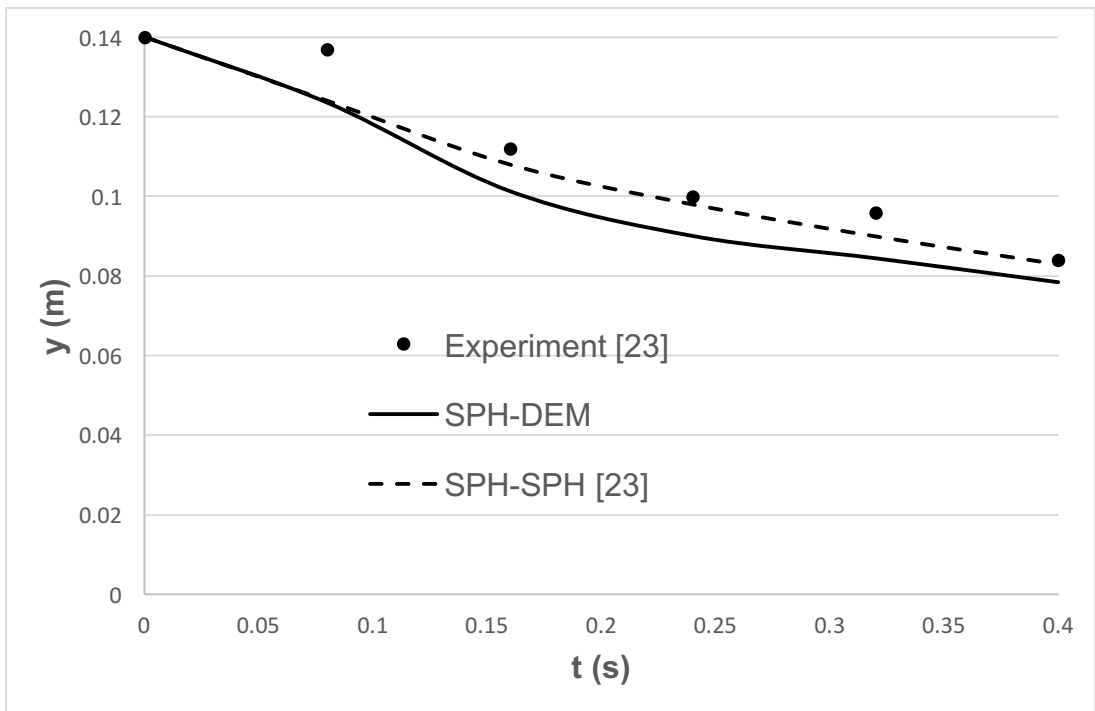


Fig. 26 Comparisons between experimental, SPH-SPH and SPH-DEM results of elastic gate test with a time period of 0.40s.



(a) Water level behind the gate



(b) Water level 5cm far from the gate

Fig. 27 Water levels at different time

In Fig.27 the water levels behind the gate and the water level 5cm far from the gate are quantitatively recorded to represent the evolution of free surface with time. Due to the water leakage in the experiment, the flow rates calculated from both SPH-SPH and SPH-DEM models are slightly higher than the experimental data, which results in a faster decrease of water level. The SPH-SPH model and SPH-DEM model show good agreement throughout the entire test process in terms of plate deformation and vertical displacement of free surface.

7.4 Dam break with bottom-fixed elastic gate

In order to demonstrate the versatility of the coupled SPH-DEM model in simulating fluid-structure interaction in this section, an FSI problem with fracture is presented. The same configuration in the validation test of coupled SPH-DEM model is used, as shown in Fig.28, but the elastic plate is clamped at the bottom end and free to move at the top end. In addition, the fixed plate in pink is removed in this case, and the material strength limit is lowered in order to allow for a fracture to occur due to the pressure of the water. Simulation parameters in Table 7 are used here. In this case, the boundary particles not only produce repulsive forces to SPH particles but also need to respond to the fractured bonded DEM particles when they contact. As the momentum energy of fractured bonded DEM particles is much greater than an individual SPH particle, the two-layer boundary particles cannot produce large enough force to impede the penetration of fractured bonded DEM particles. To solve this issue, two different kinds of repulsive forces are created from every single boundary particle to act on the

SPH particles and DEM particles, *i.e.*, the smaller repulsive forces are only designated for SPH particles while fractured bonded DEM particles receive other greater repulsive forces.

In Fig.29 (a) and (b), the largest stress (stress component σ_{22}) is found near to the bottom end of elastic plate before the occurrence of fracture due to the maximum bending moment induced by the water pressure. At time around 0.12s, the strength limit of elastic plate is exceeded and consequently the elastic plate breaks into two parts and then the fractured part moves towards the left boundary wall under the forces produced by the SPH particles of water. With the modification of the boundary particles in handling the approaching bonded DEM particles, wall penetration is fully avoided. Due to the vibration of the elastic plate, the flow pattern of water is highly affected to cause flow fluctuation which leads to some irregular movements of certain individual particles or small clusters of particles. After 0.16s, the fractured structure is pushed away to approach the left solid boundary. When the plate moves along with bottom solid wall, the stresses acting on bond are negligible as no significant deformation is observed. This coupled SPH-DEM model used in FSI with fracture is not experimentally validated yet, but these results demonstrate its capabilities.

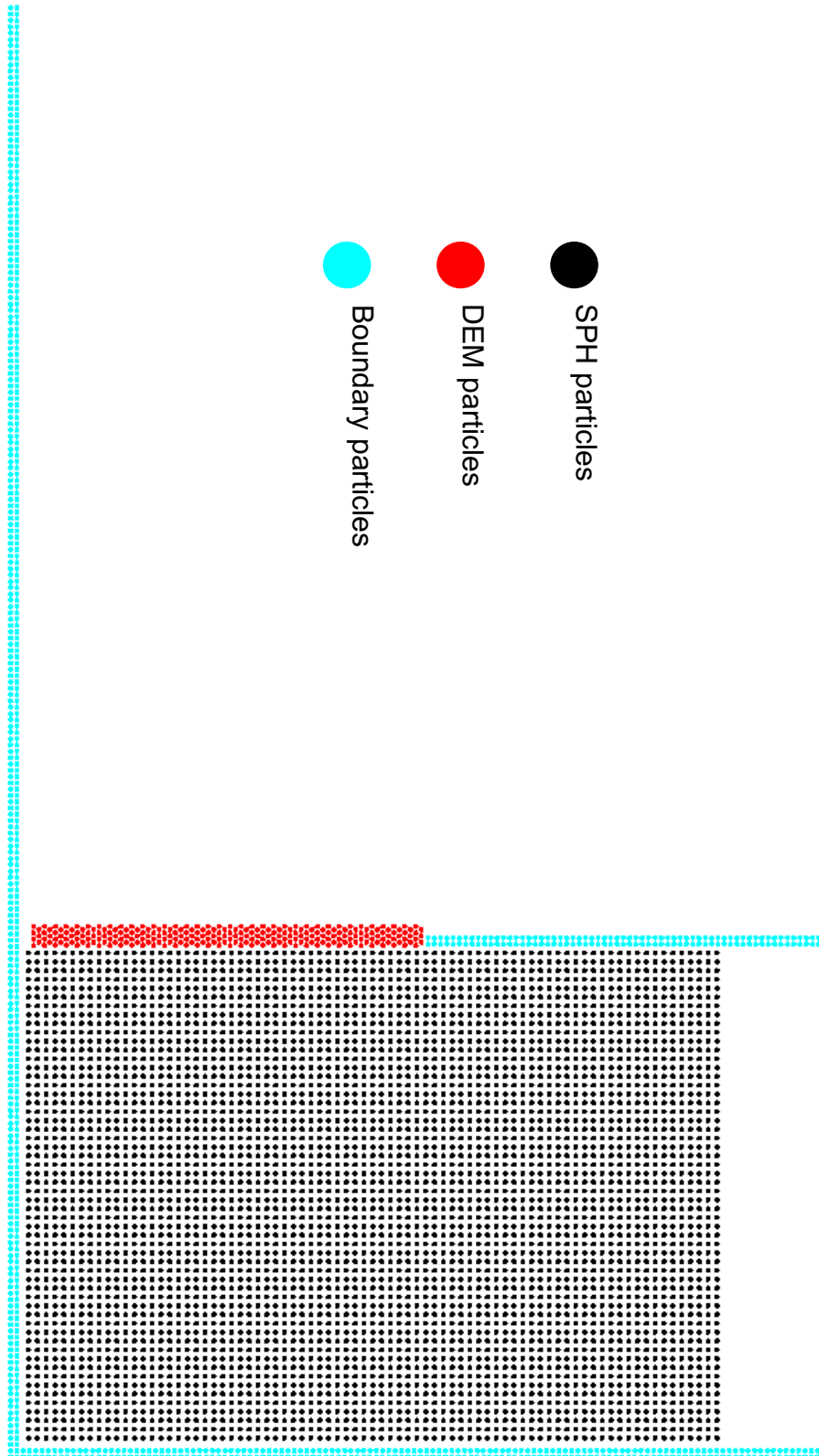
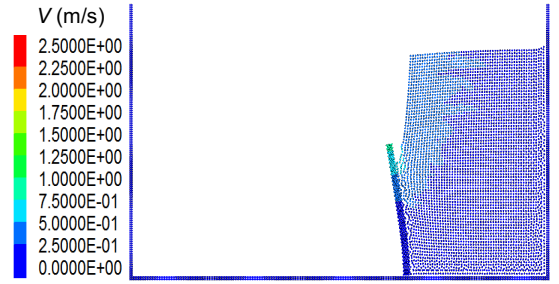
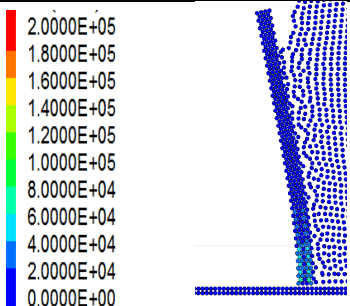
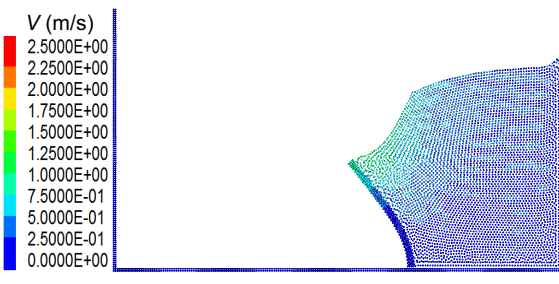
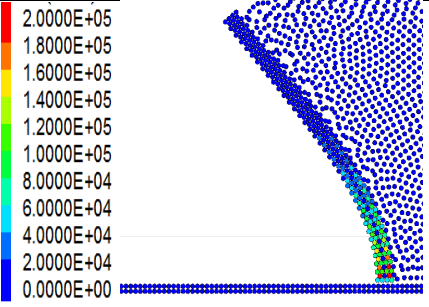
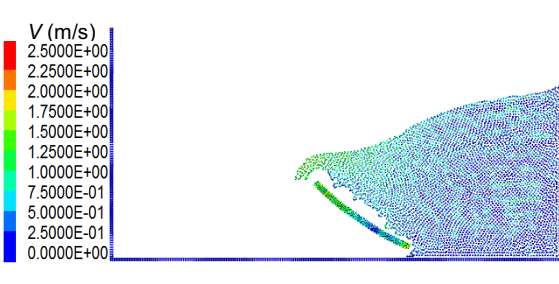
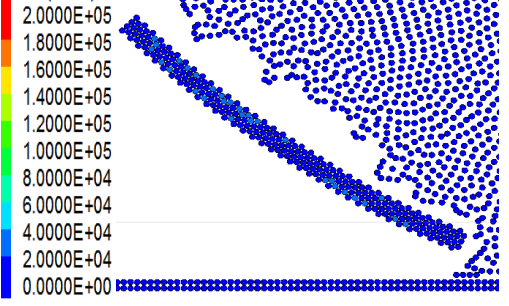
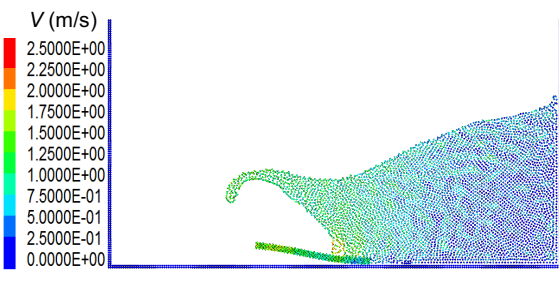
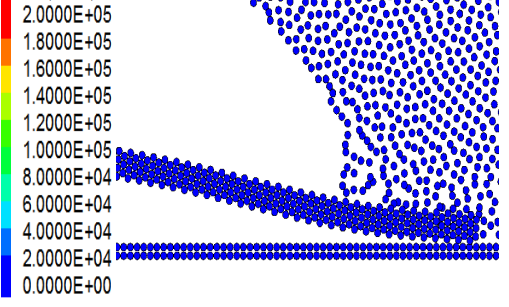
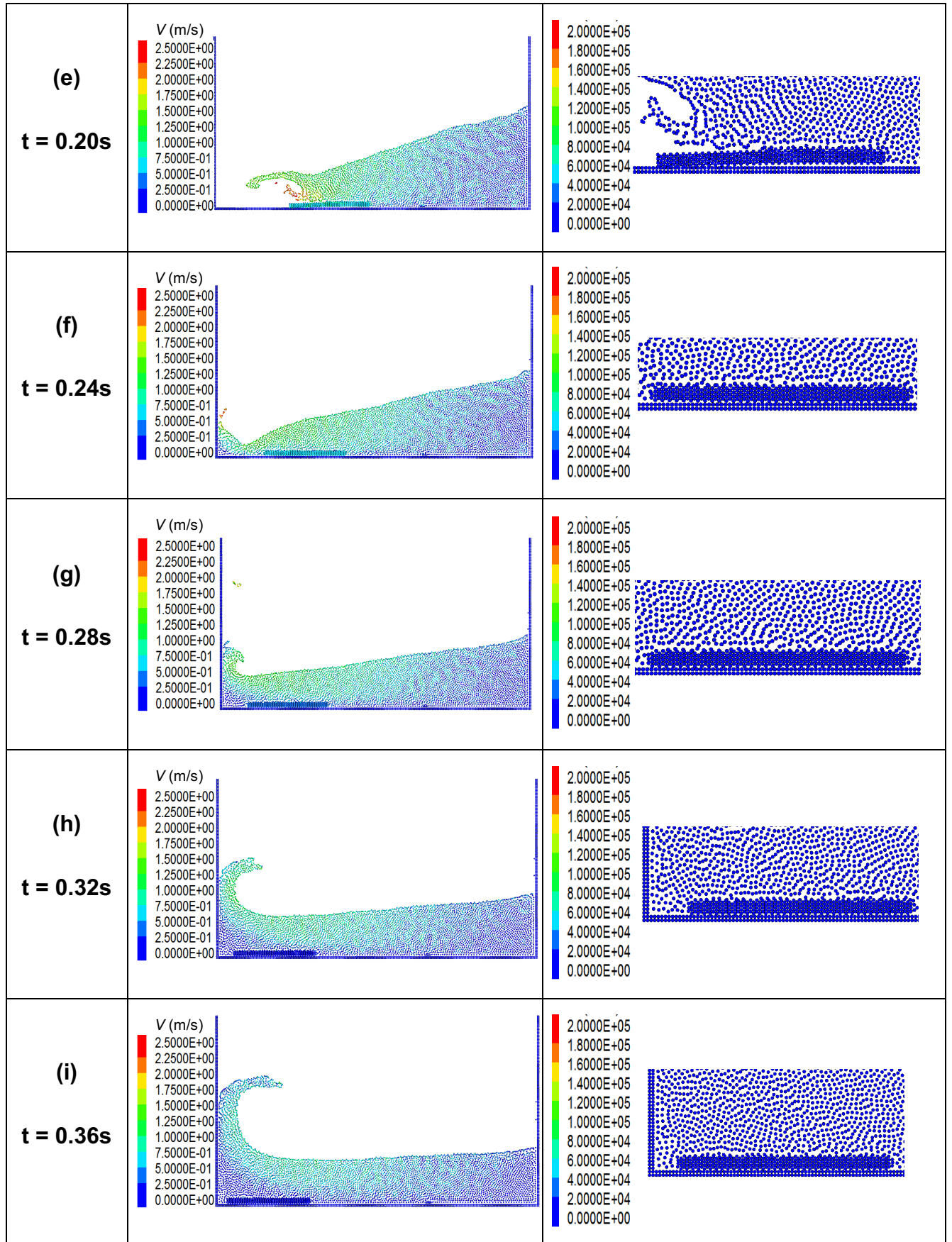


Fig. 28 2D representation of FSI with fracture

Time	Particles velocity (m/s)	σ_{22} in the structure (Pa)
(a) t = 0.04s	 <p>Velocity field at t = 0.04s. The color scale for V (m/s) ranges from 0.0000E+00 (blue) to 2.5000E+00 (red). The flow is primarily horizontal, with a slight upward deflection near the structure.</p>	 <p>Stress field at t = 0.04s. The color scale for σ_{22} (Pa) ranges from 0.0000E+00 (blue) to 2.0000E+05 (red). High stress is concentrated at the base of the structure.</p>
(b) t = 0.08s	 <p>Velocity field at t = 0.08s. The flow shows a more pronounced upward deflection and a slight expansion of the velocity profile.</p>	 <p>Stress field at t = 0.08s. The high-stress region has expanded upwards along the structure.</p>
(c) t = 0.12s	 <p>Velocity field at t = 0.12s. The flow is significantly deflected upwards and shows a more complex, turbulent-like structure.</p>	 <p>Stress field at t = 0.12s. The high-stress region has moved further up the structure, and the stress distribution is more heterogeneous.</p>
(d) t = 0.16s	 <p>Velocity field at t = 0.16s. The flow is highly deflected and shows a very complex, turbulent structure with a large wake.</p>	 <p>Stress field at t = 0.16s. The high-stress region is now concentrated in the upper part of the structure, with a significant stress gradient.</p>



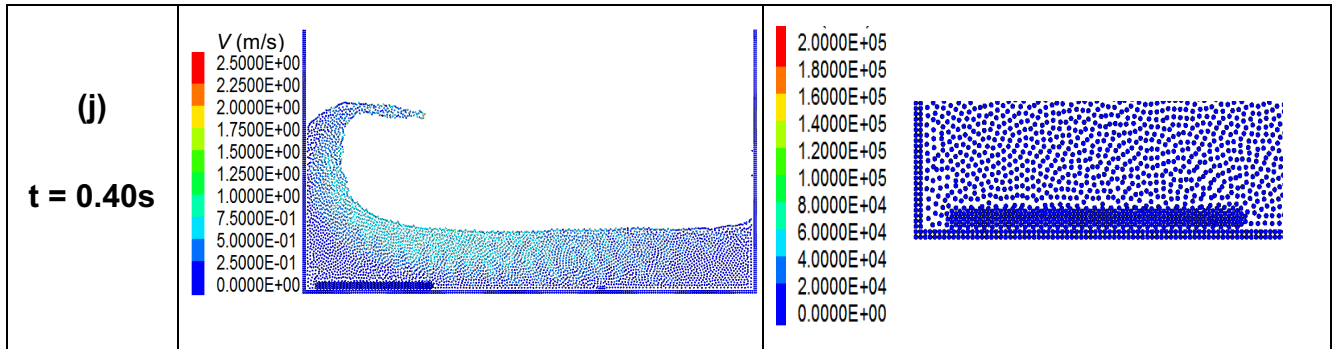


Fig. 29 SPH-DEM modelling of FSI with fracture

7.5 Two phase dam-break test with bottom-fixed elastic gate

In this section, a test including the interaction between fluid, particles and structure is simulated to demonstrate the capability of the integrated particle model to tackle the simultaneous interaction between fluid, particles and structure. Due to the direct contact between solid particles and structure particle elements, same linear contact model in DEM used in particle-particle interaction is adopted for calculating particle/structure interaction forces. The configuration of the test is shown in Fig.30, which is similar to the previous two-phase dam break test, but the moving boundary is replaced by a deformable structure with a density of 1100 kg/m^3 , which bottom is fixed. The material properties and numerical parameters for fluid and solid particles used are the same as those in multiple particles sedimentation. Two scenarios are considered by assigning different failure strengths for the structure to better illustrate the initiation of failure as well as post-failure behaviour. The tensile strength of parallel bonds is set as $4.0 \times 10^4 \text{ Pa}$ and $2.0 \times 10^4 \text{ Pa}$ in Case I and II, respectively. The contact stiffness in normal and shear directions derived through [136] are set as 1.021×10^9 and 1.024×10^7

in both cases. Relatively low strength values are deliberately chosen in order to allow the fluid induced fracture to occur.

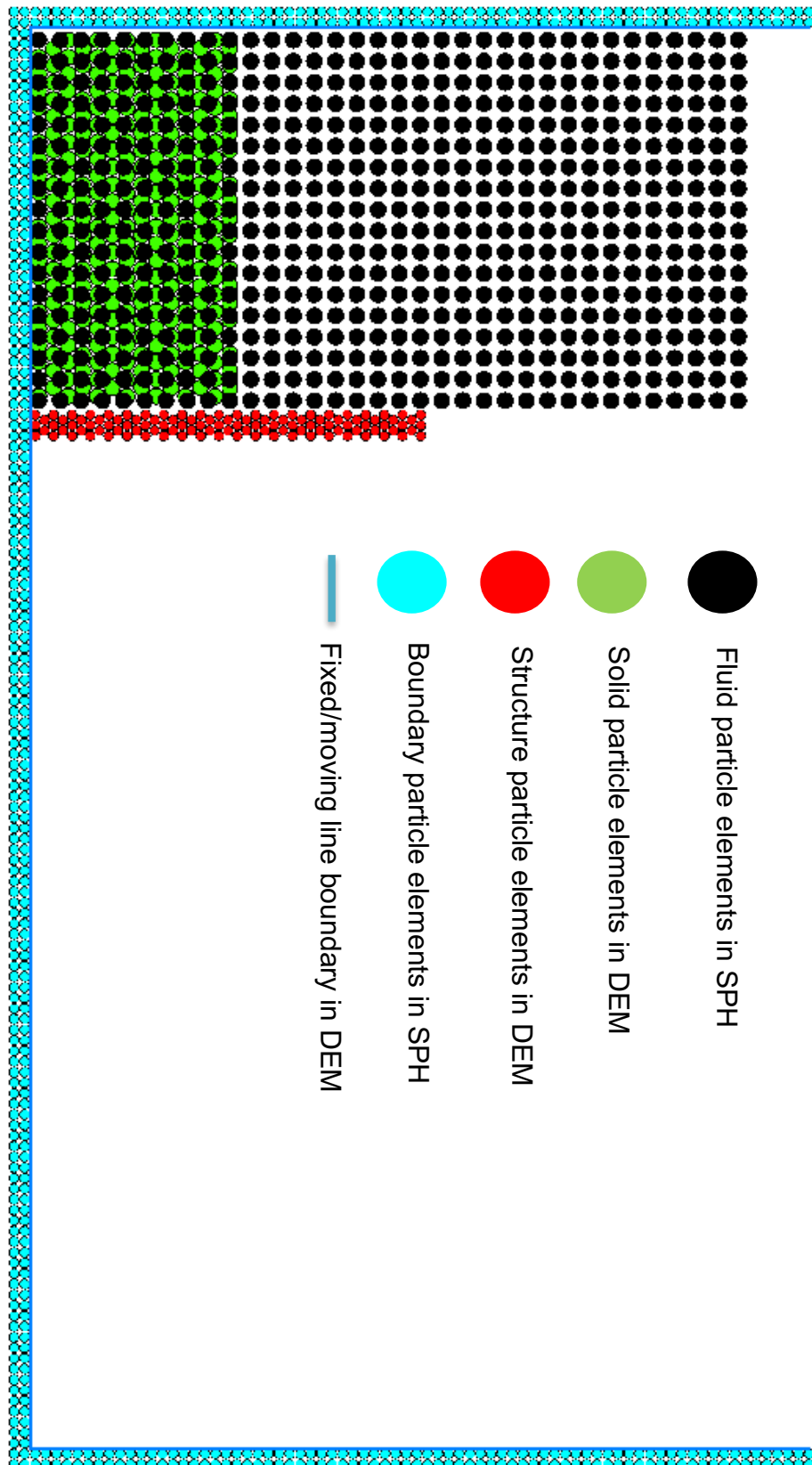


Fig. 30 Configuration of the dam-break test with Fluid-Solid particle-Structure interaction

In Fig.31, at $t = 0.05$ s, in both cases, the structure deforms due to resultant forces from the fluid and the solid particles. For visualisation purpose, the SPH particles are not plotted out and velocity vector is presented to show the fluid flow. In case II the structure has larger deformation before it fails around 0.1 s. For the structure with a lower strength, it breaks into more small pieces after hitting the bottom wall, which moves like debris and consequently makes the fluid flow more complex. It can also be clearly seen the fluid flow through the gaps between the debris. On the contrary, the structure with a higher strength has more cracks near the bottom end at $t = 0.1$ s, and the fluid tends to overpass the failed structure resulting less displacement along the bottom wall. This integrated particle model used in FSI with structural failure is not experimentally validated yet, but these results have demonstrated its capabilities.

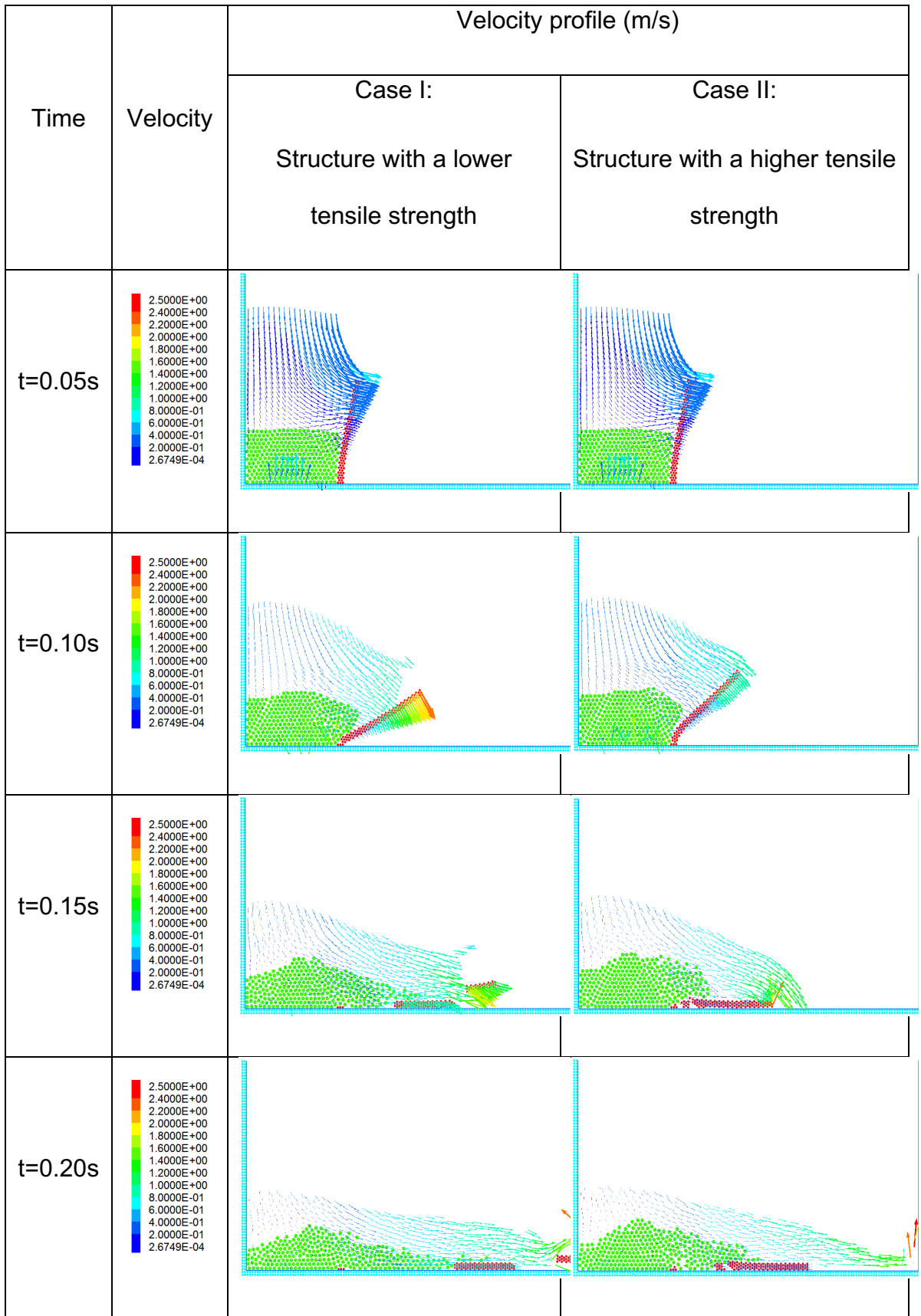


Fig. 31 SPH-DEM modelling of FSI with fracture

8 3D Printing of Fibre Reinforced Polymer Composites

Fibre-reinforced polymer (FRP) composites are extensively used in lightweight constructions in the aerospace, automotive, infrastructure, energy and sports sectors. Commercially used FRPs are mainly based on the use of thermosetting polymer matrices, but the increasing demand for rapid processes and improved impact performance has led to an increased interest in the use of thermoplastic polymer matrices [148]. Additionally, the pursuit of higher performance composites with more complex shapes and even lighter weight continues. The ever new applications of FRPs pose a significant challenge to the traditional composites manufacturing technologies which have limited control of the internal structures. Very often the traditional manufacturing of composites still requires machining/trimming/drilling and joining processes in order to meet the final geometric requirements, which further increases the risk of introducing random manufacturing defects/voids which are difficult to predict and could cause catastrophic failure.

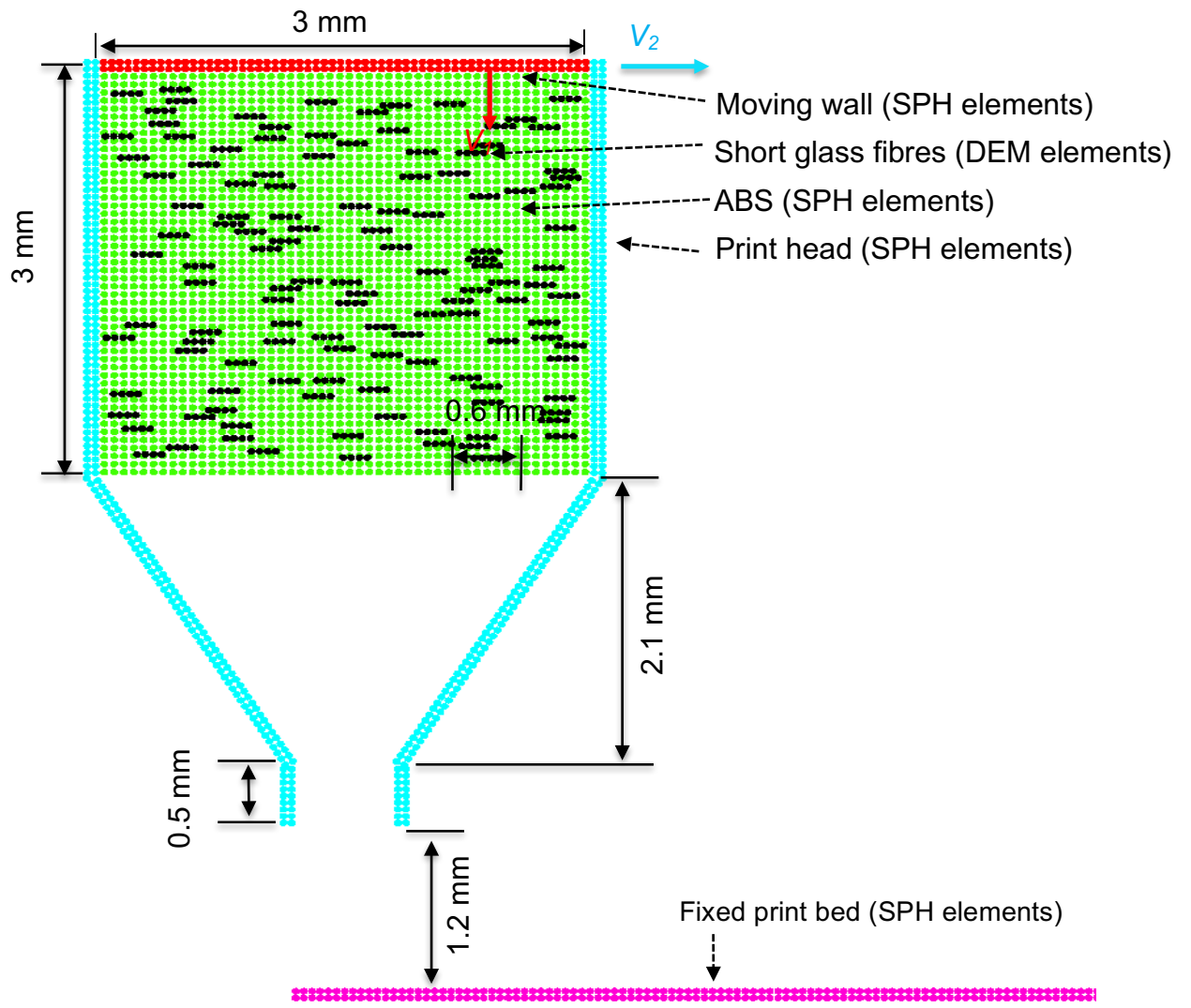
In contrast, additive manufacturing (AM) builds a part layer by layer based on CAD models. AM's fast progress in recent times has enabled material usage from initially single polymers to latterly polymer composites. This move towards the use of polymer composites for AM was driven by the fact that printed polymer (mainly thermoplastic) products show lack of strength and are limited in functionality whilst fully functional and load-bearing parts are required by industries. Incorporation of fibre reinforcement in the printing polymer leads to improved mechanical performance and additional

functionality [149]. Existing AM technologies for FRP composites are mainly based on thermal extrusion methods, such as Fused Deposition Modelling (FDM) [150] and Direct Writing (DW) [151]. In FDM, short fibres and polymer pellets are mixed and fabricated into extruded filaments for printing. Whilst in DW processes, short fibres and polymer paste are mixed and extruded directly from the printer. The Mark One and Mark Two printers recently developed by *Markforged* is capable of printing continuous carbon fibres using FDM [152]. Other different AM technologies for FRP composites are Selective Lamination Composite Object Manufacturing (SLCOM) which cuts and bonds woven composite sheets [153], and Laser Powder Bed Fusion (LPBF) which uses laser to melt a fibre-filled polymer powder [154]. In this study, we will focus on the extrusion based FDM technology. It should be noted that together with control on the printing path, FDM 3D printing provides the flexibility to incorporate multi-functionality, such as ultrasound [155] and magnetic field [156], to print composites with desired architectures. However, one of the main limitations of FDM printing is that the printed composites have lower mechanical performance compared to traditionally manufactured composites, due to fibre misalignment and porosity (or voids) introduced during the manufacturing process [149]. Most of the current research is focused mostly on investigating the effects of process and material parameters on the mechanical performance of printed part [157-159]. However, the inter-relationship between the materials, process and product is still not fully studied and incorporated. Therefore, 3D printing of composites is still a trial and error process which lacks systematic studies, *i.e.*, combination of analytic, computational and experimental approaches. There is lack of an integrated modelling and optimisation tools for the design

and implementation of 3D printing of composites. To maximize the advantages of 3D printing for composites, there is a need for the development of a computer tool to model and optimise the printing process. The FDM printing process of short fibre reinforced composites is a fluid-solid flow problem, where fibres are the solid and the melted plastic is the fluid. The FDM printing process of continuous fibre reinforced composites is a typical fluid-structure interaction (FSI) problem, where fibres deform and deposit in the melted plastic which is extruded as a fluid. In our previous work we have developed particle models for both fluid-solid flow [160] and fluid-structure interaction [161] which allows free surface flow of melted plastic, fibre-resin interaction and fibre collision/failure. It should be noted that several particle methods have been proposed and adopted to model the 3D printing of single phase materials, such as polymer [162] and metal powder [163], but to the authors' best knowledge there has not been reported process modelling of 3D printing of composites.

Thus in this study, two cases of 3D printing of composites are investigated using the coupled SPH-DEM model to demonstrate its capability and potential for simulating the printing process of short and continuous fibre reinforced polymer composites. The geometry of the print head and the printing process parameters are shown in Figure 32. The configurations for the two cases are slightly different that the height of print head chamber where initially holds resin and fibres in the case of short glass fibre is 3 mm, but 5 mm in the case of continuous carbon fibre to ensure a reasonable length of fibre are printed out, as shown in Figure 32. The widths of the print head and shower nozzle are 3 mm and 0.6 mm, respectively. All of the dimensions are the same except for the length of the print head chamber in

both cases. The short glass fiber and continuous carbon are soaked in the resin in both cases. The short glass fibers are created randomly and distributed parallel to the moving wall, while the continuous carbon fiber is located at the center of the print head, perpendicular to the moving wall. In the both cases, the moving wall, print head, fixed print bed, and resins are made of the SPH elements while the fibers are made of DEM elements. Specifically, the material of resin in the first case is ABS while Nylon for the second case. In the second case, the continuous carbon fiber is three layers wide of the DEM particles and as long as the printing head chamber. There are very limited data published for the extrusion nozzle and various feed rates and translational printing speeds are used in other research, therefore, in this study we adopt a print nozzle similar to, but ten times smaller than, those used in short fibre injection moulding. The feed rate of V_1 and printing speed of V_2 are set up in 30 mm/s and 40 mm/s respectively. In the current study, the 3D printing is assumed to be processed in a constant high temperature without any solidification. As a result, there is no energy transformation and the movement of fibre is only due to the resin flow.



(a)

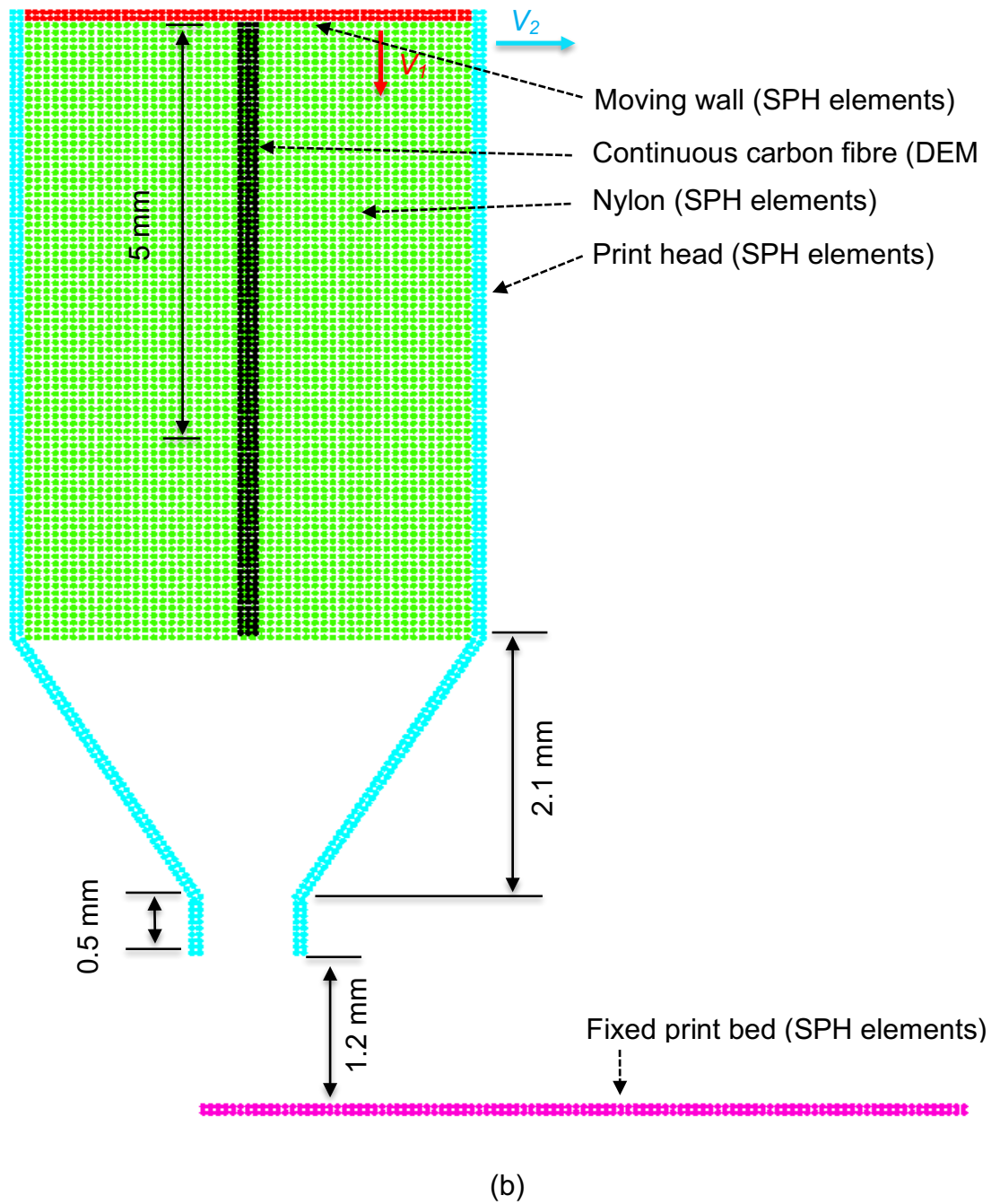


Fig. 32 Initial configurations of the 3D printing processes of composites. (a) Short glass fibre reinforced acrylonitrile–butadiene–styrene (ABS) composite; (b) Continuous carbon fibre reinforced Nylon composite.

The first case is concerned with the FDM printing of short glass fibre reinforced acrylonitrile–butadiene–styrene (ABS) composites. Each short glass fibre is represented by four DEM particles, which are bonded together using a linear parallel bond model. The collision between fibres is accounted using a linear contact model which is composed of a spring and a dashpot in both normal and tangential directions, as well as a frictional element. The density of ABS is 1050 kg/m^3 and its dynamic viscosity is $0.1 \text{ Pa}\cdot\text{s}$. The density of short glass fibre is 2540 kg/m^3 , each has a dimension of $100 \times 25 \text{ }\mu\text{m}$, and the total fibre volume fraction is 8.3% (equivalent to a weight content of 18%). The ultimate tensile strength of fibres is set as 2000 MPa.

When printing the continuous carbon fibre reinforced thermoplastic composites in FDM, usually a fabricated filament is first melted and then extruded through the printer head. The specific carbon filament for Mark Two (Markforge) consists of 1000 carbon fibres (T300, with a density of 1600 kg/m^3) and the matrix is most likely Nylon 6 (with a density of 1150 kg/m^3 and dynamic viscosity of $0.1 \text{ Pa}\cdot\text{s}$). In our current 2D simulations, it is not reasonable to model the individual fibres as they are randomly distributed in the filament, instead a nominal fibre is created to occupy the same volumetric space in the printing head, i.e., 34.5%. In this study, three layers of particles are vertically bonded together to form the nominal fibre bundle and the parallel bond stiffness is defined as follow:

$$\bar{k}_n = \frac{E}{R_a + R_b} \quad (88)$$

$$\bar{k}_s = \frac{\bar{k}_n}{k} \quad (89)$$

where E is the Young's modulus, R_a and R_b are radius of particles a and b , respectively, k is the normal-to-shear stiffness ratio. In Equations (88) and (89), the Young's modulus E is 75 GPa and the stiffness ratio k is set up to 0.25. As each individual particle size in the continuous fibre is identical all the time, R_a and R_b are equal to 0.025 mm.

In Figure 33, the predicted transient flow patterns at different time intervals are shown for both cases. For short glass fibre/ABS at time $t = 0.05$ s, it can be seen that the orientation of fibres becomes more random as they are driven by the resin flow. In the regions where the fibres are close to the boundary walls, a velocity gradient of resin flow is noticeable due to the effect of the wall, therefore the orientation of fibres in these regions changes significantly, and eventually the fibres tend to move in parallel with boundary walls with a lower velocity. On the contrary, the orientation of fibres in the middle of the printing head changes slightly, despite the direct contact between fibres and becomes aligned with the velocity profile of resin flow. At time $t = 0.1$ s resins and short glass fibres move out from printing head and they are in contact with the printing bed. However, the orientation of some fibres is less aligned with the resin flow direction, which is due to the direct contact with boundary wall (printing bed) or other fibres. In addition, the flow split occurs outside of the printing head results in a much more complex velocity contour of resin flow, which may have an effect over the orientation of fibres.

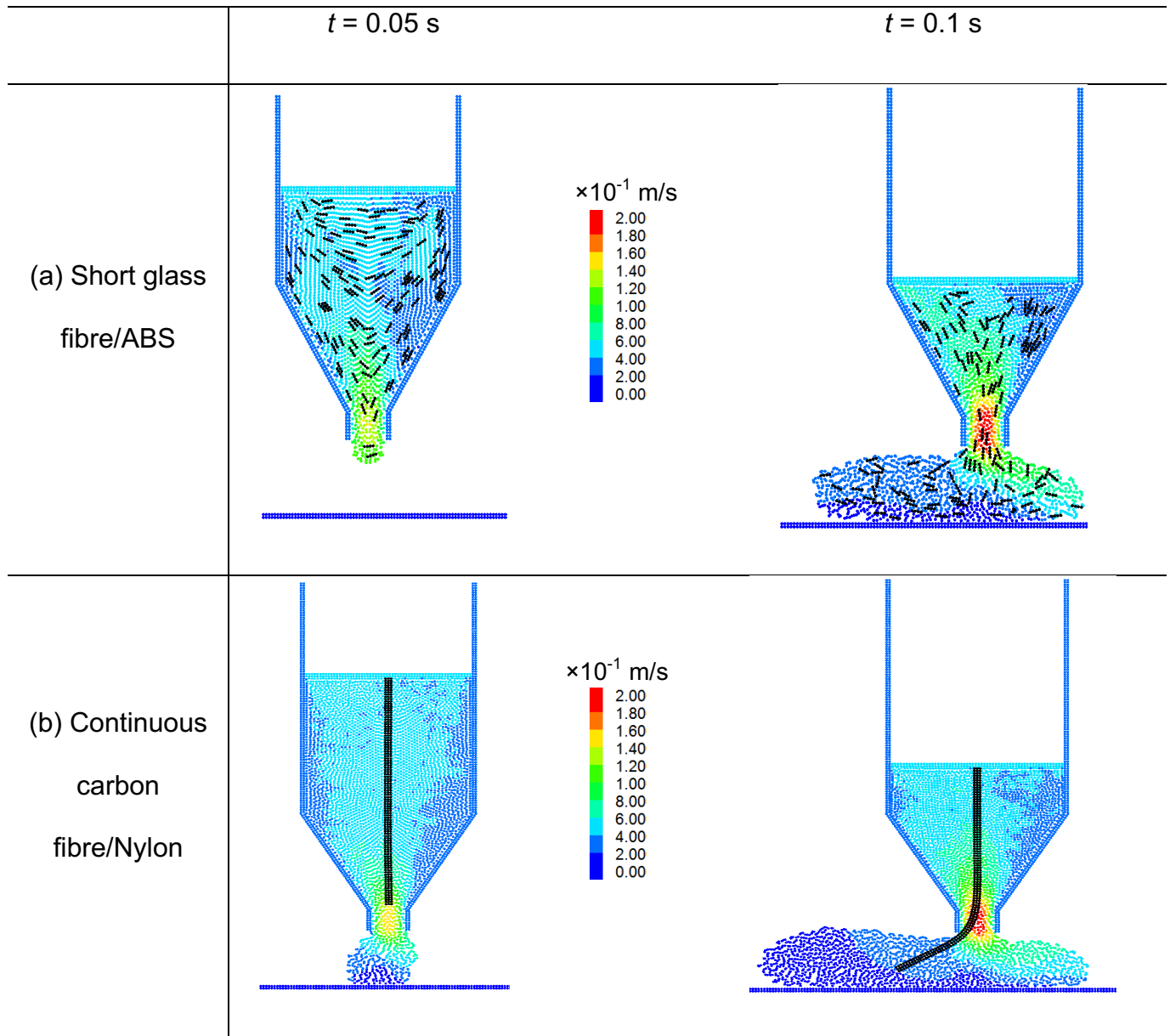


Fig. 33 Velocities of resin at printing time of $t = 0.05$ s and $t = 0.1$ s.

For continuous carbon fibre/Nylon at time $t = 0.05$ s, the fibres firmly stay in the middle of the printing head due to geometric symmetry, at least no significant vibration and deformation observed, indicating the instability effect of moving the printing head gradually reduced from the commencement of process. At time $t = 0.1$ s, the fibres show a large bending deformation and there is a continuously direct contact between fibre and left edge of the printing head, which indicates the potential wear of the printing head.

Furthermore, there are two common observations in both cases. Firstly, when resin moves out from the printing head, there are always two split streams in the fixed bed in two opposite directions. It is mainly caused by the squeezing pressure within the print nozzle. The maximum velocity of resin flow is found in the print nozzle, where the resin starts to split and drives the fibres to move closer to the inner wall of the nozzle, which indicates the potential abrasive wear as reported in experiments [28]. In the meantime, as the moving velocity of printing head is constant all of the time, consequently, the velocity of resin just outside of the print nozzle will move faster than it, but confined by the fixed printing bed due to wall effect. Secondly, the velocity profile of resin in the print head is not evenly distributed, and it is evident that the velocity of resin on the right hand side of print head is relatively lower than that it is on the left hand side. As the print head moves from left to right, the left wall of the print head passively gives a force to drive the resin, but the force is firstly transferred to the resin close to the left wall of print head, therefore, there is a delay in force transition to the resin close to the right wall of print head. In Figure 34, which shows the distribution of resin density, the density of the printed resin is lower than that under the squeezed process in the print head and print nozzle. This is due to the assumption in SPH that the incompressible flow has a compressibility feature in order to reflect the pressure distribution indirectly. In Figure 35, the deformation of continuous fibres is not visibly noticeable, and the stress is evenly distributed and not in a high order of magnitude at time $t = 0.05$ s. This is due to the fact that the fully immersed continuous fibres are still within the print head at $t = 0.05$ s, and the velocity contour in either side of the continuous fibre is almost identical, the interaction forces acting on

continuous fibre are ignorable, and as a result, no significant deformation is observed. At time $t = 0.1$ s, the tensile stress is significant due to the bending. The fibres are in contact with the lower end of left hand side of print nozzle, which could cause severe abrasive wear.

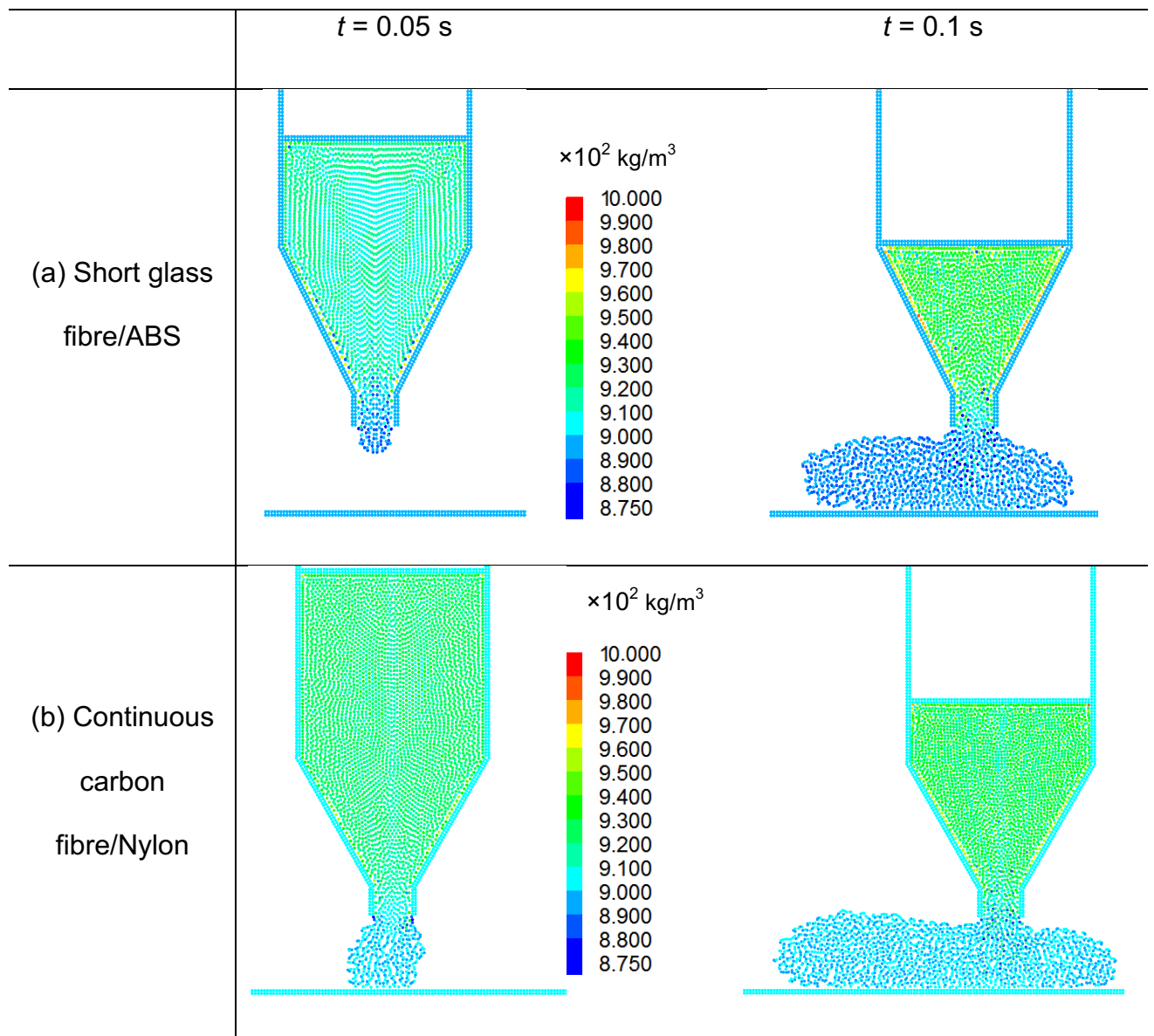


Fig. 34 Density distribution in resin at printing time of $t = 0.05$ s and $t = 0.1$ s.

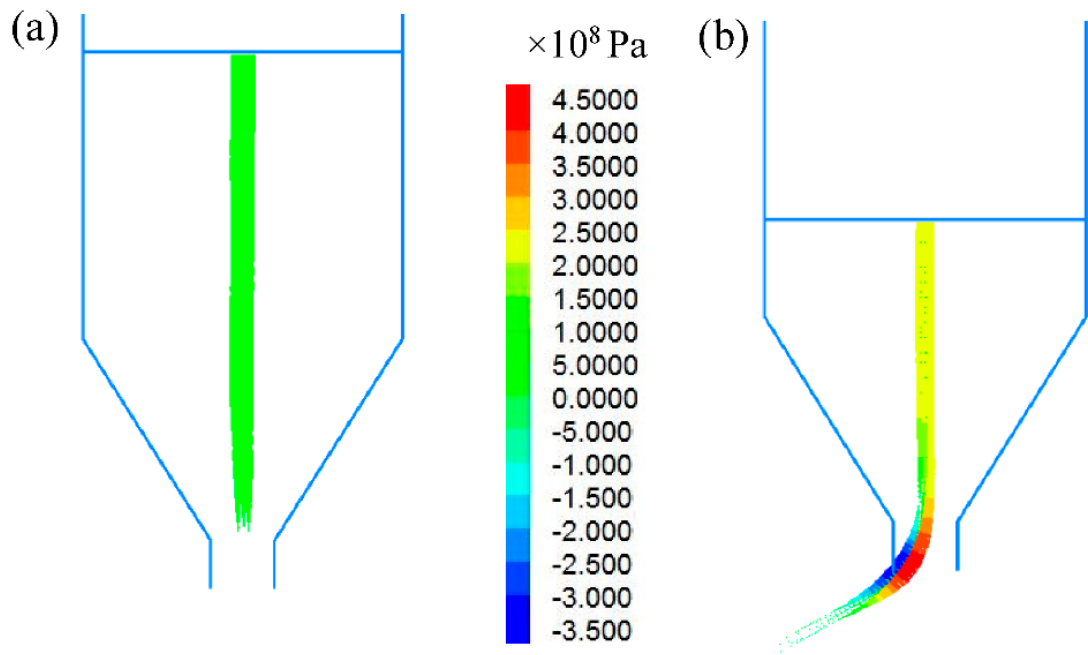


Fig. 35 Tensile stress in continuous carbon fibre/Nylon: (a) $t = 0.05 \text{ s}$ and (b) $t = 0.1 \text{ s}$.

9 2D simulation of injection moulding process of short fibre composites

Injection-moulded short-fibre composites have various kinds of advantages, such as excellent flexibility of moulding, a short time of moulding cycle and low cost of moulding as well as a certain level of stiffness and strength [164, 165]. These advantages make it very popular in automobiles and lightweight structures. The mechanical properties of these composites significantly depend on the microstructure of the fibres in the composite, such as their length, orientation and distribution. The microstructure of the injection-moulded short fibre composite is dependent on the moulding process, governed by the moulding speed, temperature and mould geometry. The orientation of reinforcing fibre has been studied since the 1980s. It was firstly investigated by Advani and Tucker [166] with an introduction of an orientation tensor to represent the orientation of fibres. They developed equations for the change of orientation tensors by combining the equation for the fibre movement with the continuity equation for all the fibres in the material. Since then, the orientation tensors are widely utilized for predicting the microstructure of the injected moulding product [167-169]. Meanwhile, based on the orientation tensor and fluid dynamics, numerical approaches have been developed to analyse the fibre motion. However, these fibre orientation-based approaches are not capable of presenting the microstructure of the composite in details as the tensors cannot predict the motion of an individual fibre. Sun *et al.* [170] and Oumer *et al.* [171] simulated the injection moulding process using a Computational Fluid

Dynamics (CFD) code, MoldFlow. Numerical results were validated against experimental observations showing that the density of fibres aligned to the flow direction in the vicinity of the mould cavity walls is large within the thin plate and very few in the thickness direction. Modhaffar *et al.* [172] developed a CFD model to simulate and characterize the fibre suspension in the flow within rectangular cavities. The model was solved with finite volume method (FVM) and it was capable of describing the temperature profile and predicting the fibre orientation during the filling stage of injection moulding process. Thi *et al.* [173] predicted the fibre orientation in the short-glass fibre composites with different fibre weight concentration during the injection moulding process using a numerical approach based on Folgar and Tucker equation [174], which is well known for modelling the fibre orientation, and the Jeffrey's equation, which aims for the fibre-fibre interaction. The numerical results were validated by the experimental observations conducted with a high resolution 3D X-ray computed tomography, quantitatively.

Better understanding of motion behaviour of fibres in resin flow can provide insights for explaining some complicated phenomenon including the accumulation of fibres within resin flow. Yamamoto and Matsuoka [175, 176] proposed a particle method named particle simulation method (PSM) to simulate the motion of fibres, in which all the fibers were modelled as an assembly of particles, and the equations of motion for each particle were solved separately. Although this approach could represent the accumulation and deformation of fibers, the resin flow and fiber motion were analyzed separately, thus the influences of fibre orientation and motion on the flow

field could not be represented. Yashiro *et al.* [164, 165] utilized the moving particle semi-implicit (MPS) method [49, 177] to simulate the mould-filling process of short-fibre composites. This method has an ability to track the free surface flow such as flow-fronts and the motion of each fibre which is represented by an assembly of particles. However, the fibres in their studies are assumed to be rigid, unlike the experimental observations that fibres could deform and even break up during the filling process [178]. In addition, both the interaction between the fibres and interaction between fibres and resin are mathematically incorporated in the MPS algorithm, without a physical interpretation. For example, the interaction between fibres due to collision is handled by indirect contact due to the nature of MPS. What is more, as both resin and fibres are represented by two types of particles in MPS scheme without any overlap, when the volume fraction of fibres is very large there would be insufficient amount of particles to accurately capture the flow of resin.

Therefore, in this study the coupled SPH-DEM approach for simulating the injection process of short fiber composites was carried out. The fibres are represented by bonded DEM particles and the resin is represented by discrete SPH particles. This coupled SPH-DEM is a meshless method in a *Lagrangian* scheme, thus mesh generation and re-meshing is not required. The approach has been comprehensively validated in [160, 179] for modelling general interactions between fluids, solid particles and structures. For simplicity and straightforward implementation, the injection moulding process of short fibre composites is carried out in a 2D numerical simulation with proposed SPH-DEM model, where gravitational acceleration is not included. The injection moulding process is often the key step to determine

the quality and material property of moulding materials. For comparison the same injection process simulated by MPS in [165] is adopted. The configuration of the mould is depicted in Fig.36. The resin is considered as an incompressible viscous fluid and represented by discrete SPH particles, and the short fibres are made of four bonded DEM particles. These short fibres initially are randomly distributed in a 30mm × 30mm square resin bath in the direction vertical to the injection direction with a volume fraction of 3.8%, hence the total quantity of short fibres is 175, same as number of the fibres used in [165]. A rigid wall at the top of the resin bath moves downwards with an initial velocity of 1 m/s to push resin and short fibre composites into a nozzle area. The velocity of the moving rigid wall is gradually decreased to ensure a stable injection moulding process. Through a stable injection, the resin and short fibres pass a narrow gate in width of 6mm and length of 5mm to completely fill up a slim plate mould with 80 mm in width and 4 mm in length. The density of resin is 900 kg/m³ and its dynamic viscosity is 0.1Pa · s. The density of short fibres is 2540 kg/m³ and every individual has a dimension of 1mm × 0.25mm. The critical ultimate tensile strength of fibres is 2000 MPa. As the aspect ratio of fiber is only 4 and the majority of materials moves along with resin flow, the interaction between resin and short fibres is insignificant and consequently the phenomenon of visible deformation of short fibres is hardly observed. However, the stress of bond in each short fibres can be extracted for quantitative analysis where necessary. The numerical parameters are listed in Table 8.

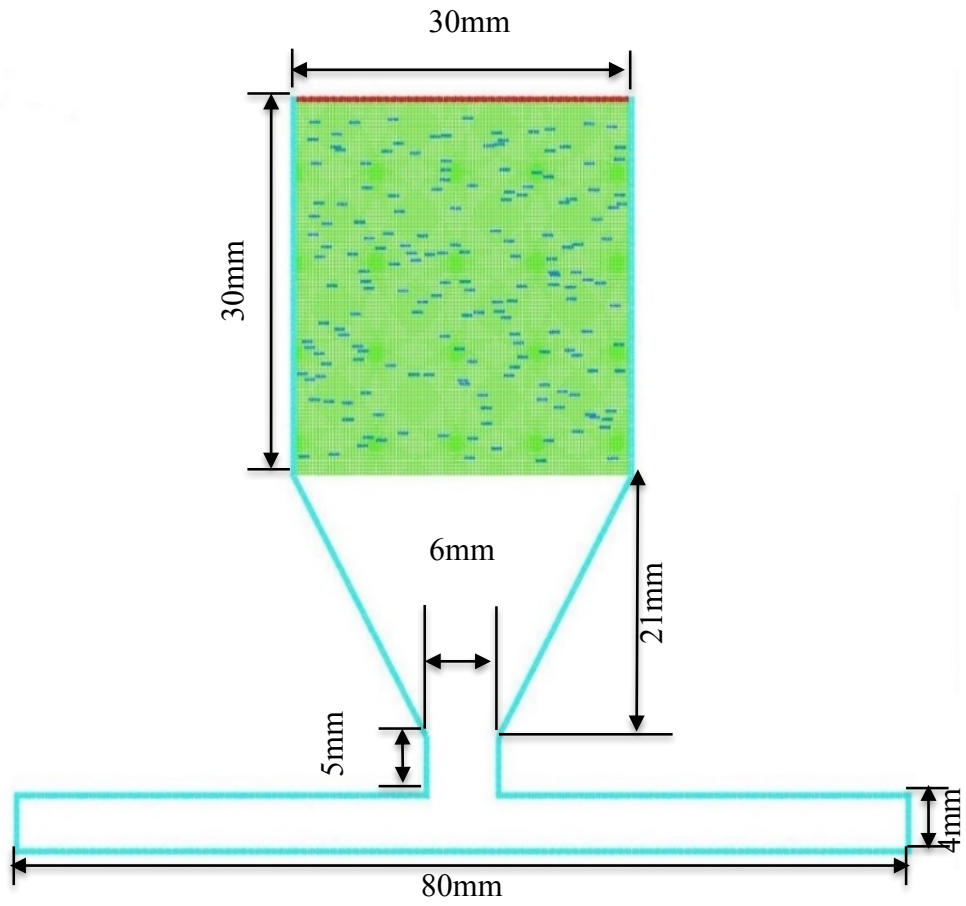


Fig. 36 Configuration of 2D SPH-DEM simulation of the injection moulding process

Table 8 Numerical parameters in SPH-DEM modelling of injection moulding process

Parameters	Values
Resin particle spacing (m)	0.0003
Kernel function	Wendland
Kernel smooth length (m)	0.00036
Time step (s)	0.0000002
Physical time (s)	0.12

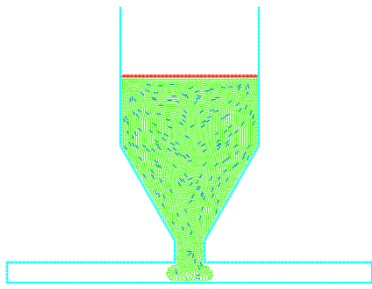
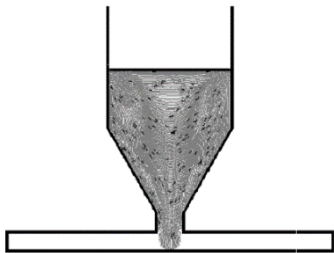
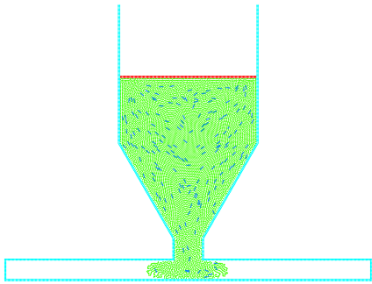
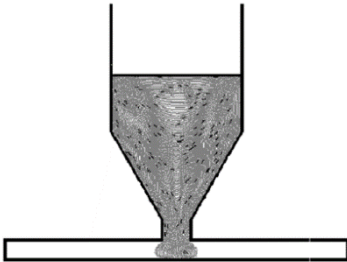
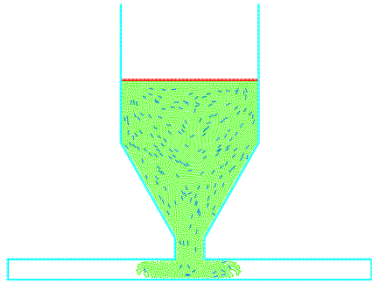
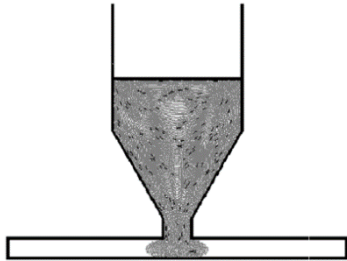
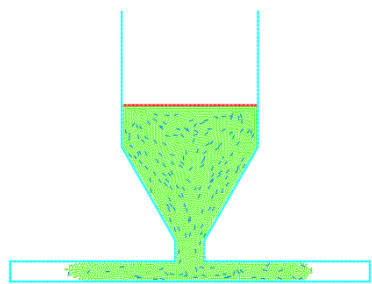
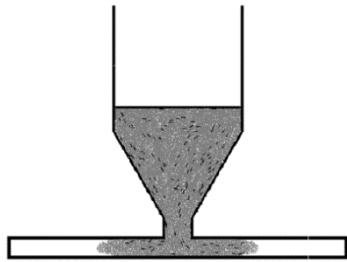
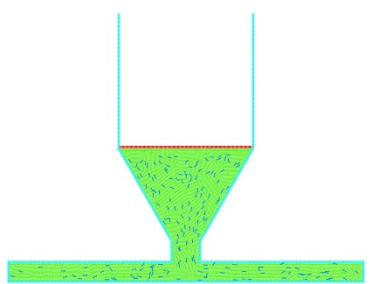
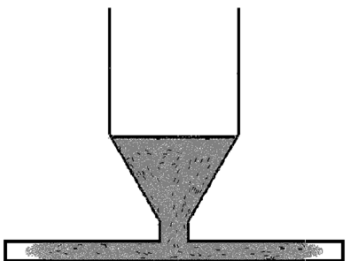
Time (s)	SPH-DEM	MPS [165]
0.02		
0.025		
0.03		
0.06		
0.12		

Fig. 37 Snapshots of the injection moulding process at different time

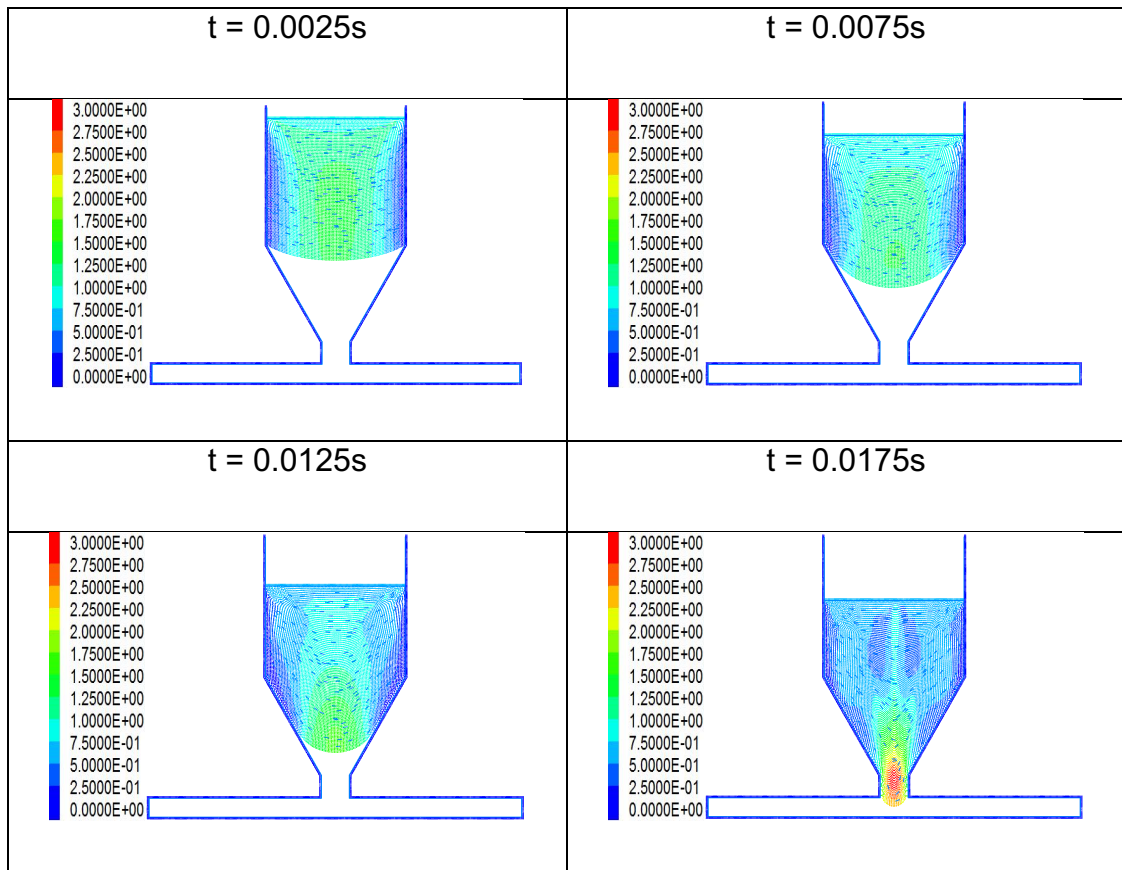


Fig. 38 Velocity contour of resin flow before split flow at a 0.005s time interval

In Fig.37, the snapshots of transient injection flow pattern at different time intervals are predicted by SPH-DEM model and compared with MPS results. The resin passes the narrow gate and starts to fill the plate mold around 0.02s. It can be seen that the orientation of fibres becomes more random as they are driven by the resin flow. In general SPH-DEM predicts similar flow profile as the MPS method. In the regions where the fibres are close to the boundary walls, a velocity gradient of resin flow is noticeable due to wall effect (see Fig.38), therefore the orientation of fibres in these regions changes significantly and eventually the fibres tend to move in parallel with boundary walls with a lower velocity. On the contrary, the orientation of fibres in the middle of the mould initially changes slightly despite of the direct

contact between fibres and gradually becomes aligned with resin flow. At time 0.0175s, two small regions in the middle of the mold are found with low velocity, this is the result of the appearances of vortex produced by the moving wall when it squeezes the resin flow in the nozzle area. After passing the narrow gate from 0.02s to 0.025s, the resin flow and fibres start to split into two streams to fill the slim plate. Similarly, the orientation of majority of fibres from 0.06s to 0.12s is still aligned with the resin flow direction, even though some of them have direct contact with boundary wall or other fibres.

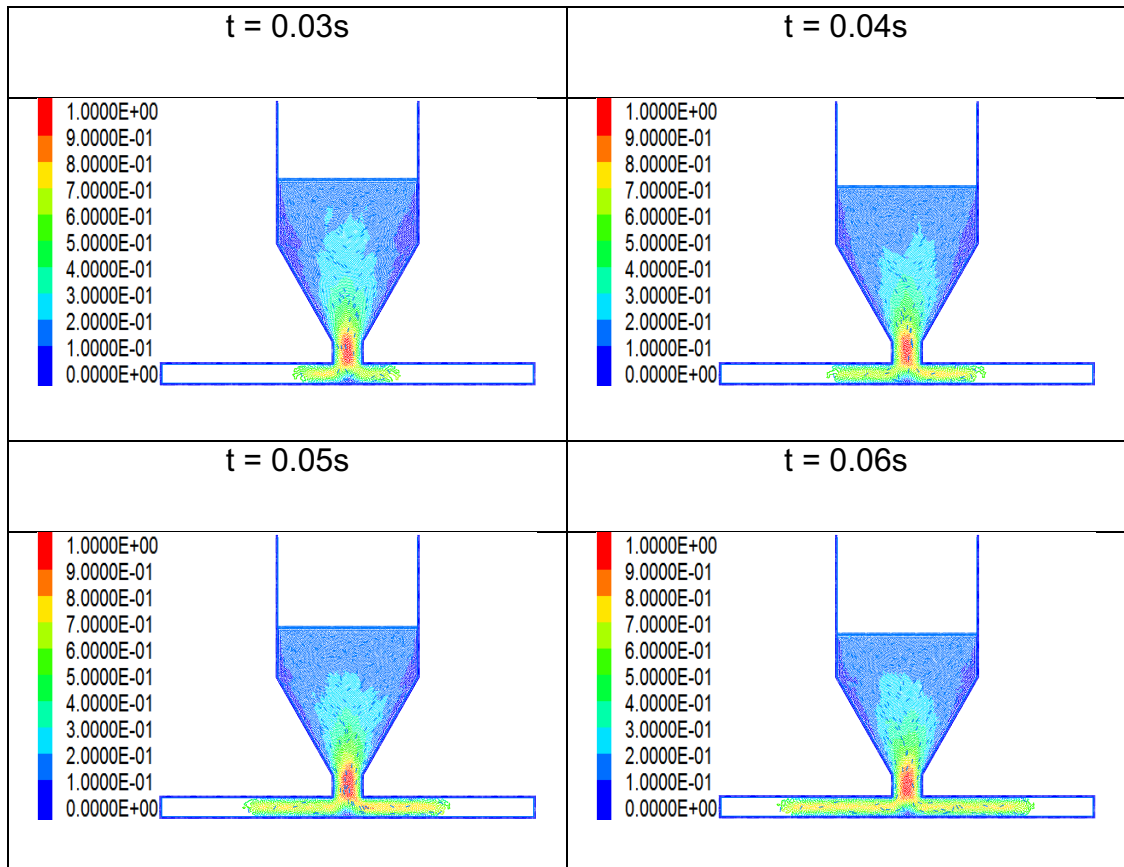


Fig. 39 Velocity contour of resin flow after split flow at 0.01s time interval

In Fig.39, it can be clearly seen that the velocity gradient of resin flow caused by wall effect is much steeper, and some fibres collide with the mold

walls resulting in random orientation. Besides, a stagnation region, where the resin flow is almost at rest, is observed in the middle of the bottom plate mould due to flow split. Finally the injection process is completed at time 0.12s, but filling process still continues until 0.165s in the MPS simulation [165]. The difference is partly caused by the velocity of moving wall, which is controllable by resin pressure at every timestep in the MPS simulation. As the threshold value of resin pressure was not given in the reference, the velocity of moving wall in the current SPH-DEM simulation is estimated to linearly decrease with an deceleration of 25 m/s^2 over the first 0.02s and then become a constant velocity of 0.15 m/s afterwards in accordance with the displacement of moving wall in each snapshots from [165] as shown in Fig.37. In addition, a slightly irregular parabolic profile of fountain flow in SPH particles is depicted in Fig.38, where an unrealistic free surface and voids in the resin flow appears in both directions at the flow front after flow split. This may be due to tensile instability and free surface tension. Although the flow front is satisfactorily in parabolic shape, it occupies more spaces in the plate mold leading to a faster filling process. This issue could be mitigated by considering the surface tension as an external force in the momentum equation of SPH, which is not investigated in this study.

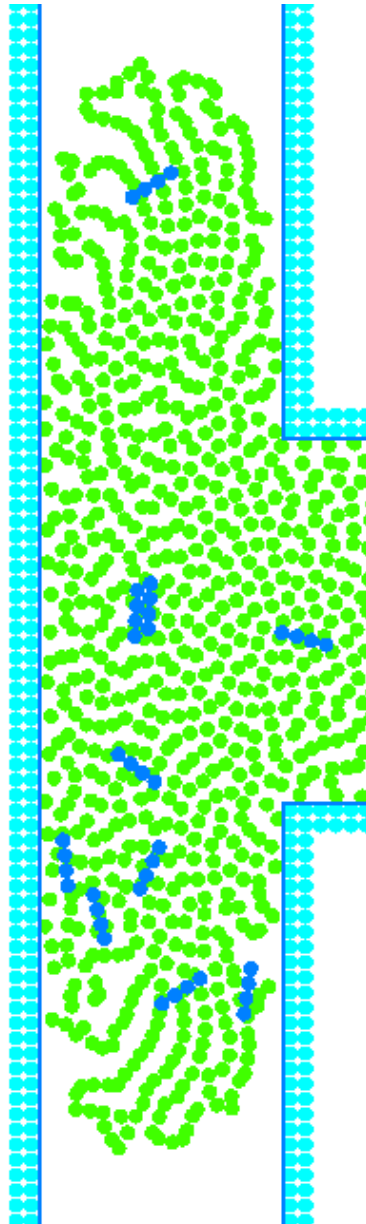


Fig. 40 Front flow observed at time 0.025s

Fig.41 plots the tensile stress of parallel bond in each fibres at time 0.2s. It can be seen that the maximum tensile stress 936.51MPa is far below the critical tensile strength of fiber (2000 MPa). This is due to the fact that the short fibres are mainly driven by shear flow to move with the resin, the interaction force acting on the short fibres are insignificant and not enough to break the bonds. Furthermore, as the aspect ratio of fiber is only 4, the deformation of fiber cannot be noticeably observed and the tensile stress caused by tension and bending in each fibre is almost evenly distributed. It should be noted the shear stresses in each fibre are also not noticeable and there is little deformation perpendicular to the fibre axis.

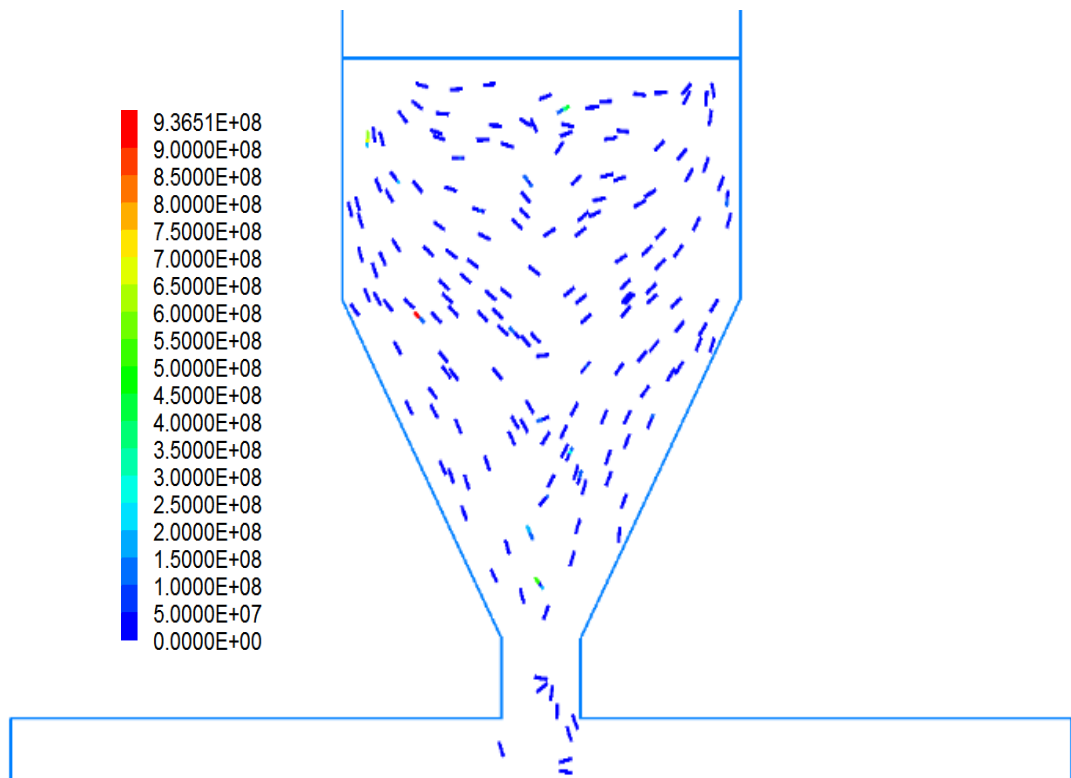
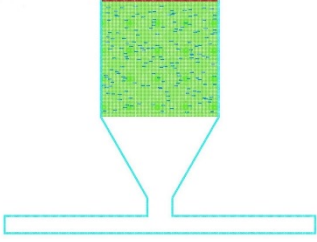
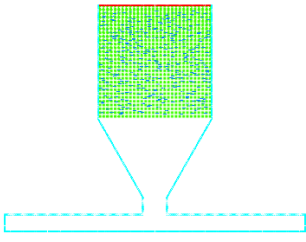
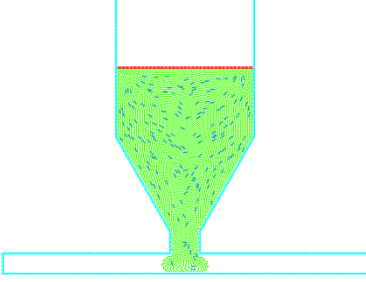
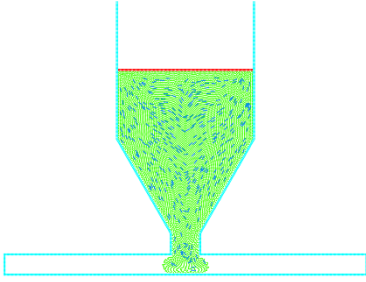
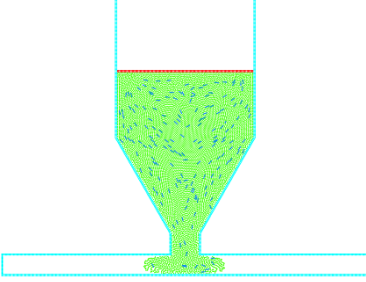
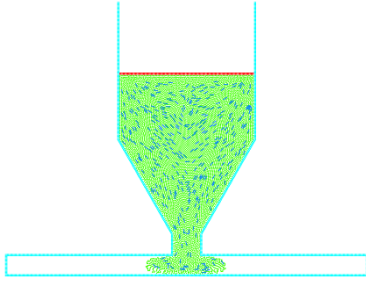
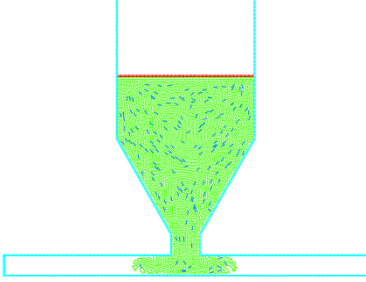
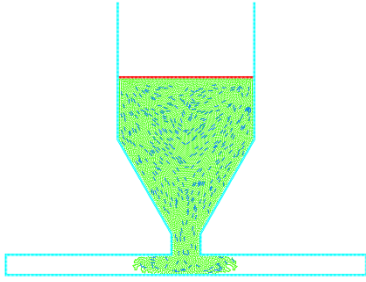
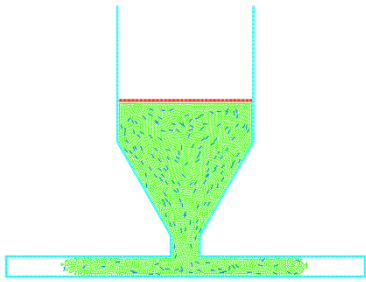
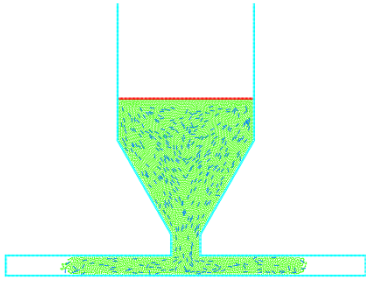


Fig. 41 Distribution of tensile stresses in short fibers at time 0.2s

Another simulation with a high fibre volume fraction of 8.3%, shown in Fig.42, is conducted in order to compare with experimental results in [164] and investigate the influence of fibre contents on fibre orientation and distribution. The total quantity of short fibres is 375. The dynamic behaviour of resin flow and short fibers is similar to the previous case, where the fibres close to the skin layer of mould are almost aligned with the resin flow direction and the fibres in the core of mould are in random orientation. However, more accumulation of fibres is found as shown in Fig.42. Since the resin flow velocity is larger at a distance far from the boundary wall due to the viscosity of the resin, more frequent collision between fibers occurs in comparison to the case with volume fraction of 3.8% and this partly leads to a more random orientation of short fibres in despite of resin flow, even though the fibre orientation is mainly governed by the shear rate [166]. For these fibres close to boundary wall, their velocity is relatively lower due to the boundary effect which results in more accumulation. Consequently composite with higher fibre content is more likely to have more disrupted orientation and irregular distributions of fibres after injection. What is more, for simulation with a high fibre volume fraction of 8.3%, the content of fibres in the right path (+y) and left path (-y) after flow split is almost the same, but slightly more fibres are observed in the right path (+y) for simulation with a low fibre volume fraction of 3.8%. There are many factors can influence the content of fibres in either paths of mold. It can be controlled by adjusting the distribution and volume fraction of short fibre at initial stage and the velocity of moving wall (e.g. constant or variable velocity) to have an impact over the entire movement of short fibres during the injection process.

Time (s)	Volume fraction 3.8%	Volume fraction 8.3%
0.00		
0.02		
0.025		
0.03		
0.06		

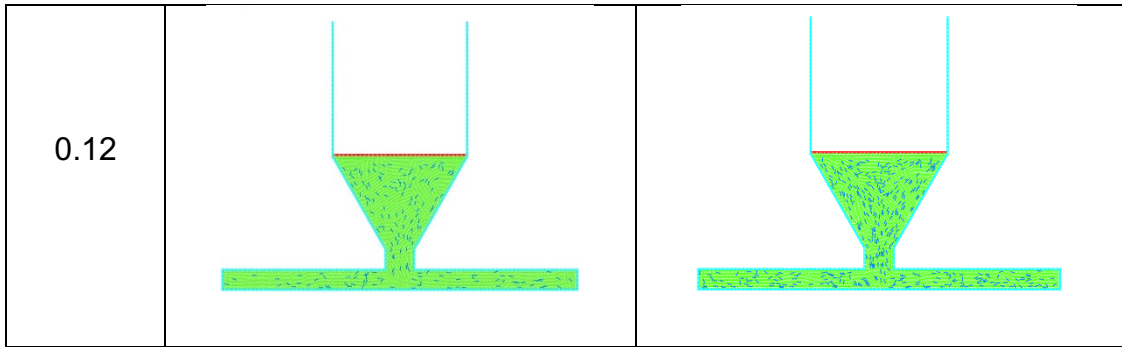


Fig. 42 Injection moulding processes with fibre volume fractions of 3.8% and 8.3%.

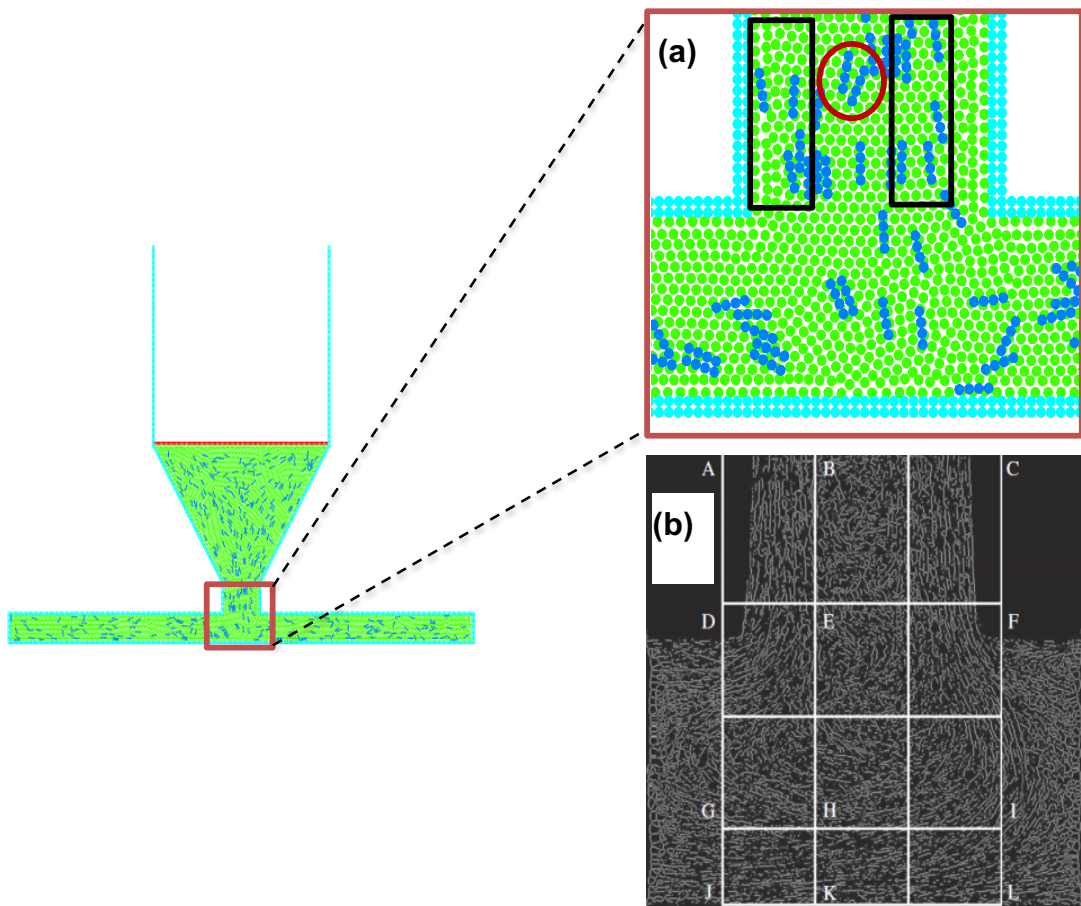


Fig. 43 Qualitative comparison between (a) numerical prediction and (b) post-processed X-ray CT image of fibre orientation upon the completion of filling process

In Fig.43, due to the limited number of fibres and relatively dense fibres in the narrow gate in the current simulation, only the upper part of the area marked by red square is extracted to compare the experimental results with areas A, B and C when the filling process is completed at 0.12s. In the author's simulation, the aspect ratio of short fiber composite is 4, nevertheless the aspect ratio in the experiment was in a range between 50 and 100. In despite of the difference in aspect ratio, Jeffery [180] demonstrated that the influence of the small aspect ratio on the orientation distribution would be limited as the change in orientation for an aspect ratio of 4 is almost the same as that for a larger aspect ratio larger than 20. Therefore, the aspect ratio of 4 can be used to similarly represent the movement of long fiber composites. In the experiment, most short fiber composites in the narrow gate area (areas A and C) moved aligning with resin flow direction, but those in the middle part of narrow gate (area B) moved randomly in any orientation and the degree of disorder is increasingly noticeable from area E to area K. As few number of short fiber composites in the narrow gate in 2D simulation, the predicted orientation of short fibers can only be qualitatively in comparison with experimental results obtained from X-ray CT. In the zoom-in view of the red square region, two short fibers marked in a red circle are mostly in disorder with maximum orientation angle compared with neighboring short fibers marked in black squares. The maximum orientation angle can be caused by either collision between particles or least velocity gradient in the middle part. It can be seen that these predictions qualitatively agree with experimental results. To more quantitatively compare with experiments, a series of 2D simulations with different initial conditions of resin bath could be carried out as in [164], and

this would take in account the unpredictable influence of short fibres distributions over the fibre orientation and movement (e.g. short fibres are not evenly distributed at the start of injection process). On the other hand, 3D simulations of the injection process would more accurately capture the details of fibre flow and orientation, but at a much higher computational expense.

10 Conclusions and future works

10.1 Conclusions

A 2D coupled SPH-DEM model for fluid-solid interaction problems has been proposed and developed. In this model, SPH based on Navier-Stokes equations is used to model the fluid phase while DEM with/without a bond feature is used for the structure and solid particle phase. In dealing with the fluid-structure interface, Newton's Third Law is applied with the same magnitude in both phases, but in opposite directions. When dealing with the interaction between the fluid and solid particles, the local averaging technique is used to account for the volume of solid particles in the fluid. As both SPH and DEM are Lagrangian particle methods, no special treatment is required to define the fluid-structure interface, even in the presence of large deformation and/or fracture of structure. The contact between SPH and DEM particles is automatically detected in accordance with a particle search radius that is twice the smooth kernel length. When a smoothed particle is approaching a fixed boundary and its support domain is intersected with the boundary, two-layer boundary particles are placed in the position of solid boundary to produce repulsive forces to handle the kernel truncation.

The individual DEM model and SPH model has been validated by comparison with analytical, experimental and other numerical results. A tip-loaded cantilever beam has been chosen to validate the DEM model with bonded particles for predicting structural deformation and fracture, whilst solid particle phase has been studied and validated in this research using a dry dam-break test with a stack of solid cylinders in two-dimensions.. A

typical dam-break test with dry bed was considered to validate the SPH model for predicting free surface flow. After the validations of both DEM and SPH models, the coupled SPH-DEM model was then validated against a typical fluid-structure/solid particle interaction problems, such as a thin and long elastic plate interacts with free surface flowing fluid, the occurrence of structural fracture by allowing the bonds in the DEM model to break and the broken structure to move under fluid pressure, a sedimentation test of a single particle and two-phase flow dam-break test. For single particle sedimentation, the fluctuation of settling velocity of a solid particle is due to the assumption that fluid is compressible in SPH theory so that the surrounding fluid particles can be compressed or expanded at any timestep, which gives rise to the fluctuation of surrounding fluid velocity and the terminal velocity of the solid particle is affected by the ratio of the resolution of the fluid particle to the diameter of the solid particle. According to authors' experience, even though the results in single particle sedimentation are satisfactory, some improvements are still needed in order achieve more accurate results as produced by other methods such as LBM. In the two-phase dam-break test, the results for the dynamic behaviour of front wave are promising, but the lack of a third dimension neglects the effect of the thickness of solid particles. The last step in the validation is to combine all phases together and a special case is presented to illustrate the fluid-solid particle-structure interaction with/without structural failure. In comparison with other results, the results obtained here are found to be satisfactory and encouraging for future work.

And then the research chose two engineering problems to show the capability of the proposed coupled SPH-DEM model. First of all, this study

coupled SPH-DEM model is applied to simulate the 3D printing process of short or continuous fibre reinforced polymer composites by fused deposition modelling (FDM). Fibres are represented by discrete DEM particles bonded together to capture the deformation and even fracture when it necessarily occurs, therefore, this model can track the motion and deformation of fibres and analyse their interaction with resin during the 3D printing process. Resin as a fluid phase is represented by discrete SPH particles. Fibre orientation and deformation are predicted to shed a light on the 3D printing of short or continuous fibre reinforced polymer composites (e.g., void formation, fibre distribution, etc.). Secondly, the simulation of injection moulding process of short fiber composites was implemented. Two cases with different volume fractions of short fibres are conducted and numerical results of the resin flow velocity and fibre orientation and accumulation are analyzed and compared to other numerical and experimental results. It's confident that the proposed SPH-DEM model can reproduce the realistic movement of fibres in the resin flow for moulding injection, which are very difficult to achieve by the conventional methods.

Finally, the original findings and contributions in the present investigation can be generally described as follow:

1. The coupled SPH-DEM model is implemented into PFC5.0 with the addition of developed SPH code, the quality of which is proved by the solution of standard tests. This model only open for fluid dynamics may be considered as the first step in the development of an advanced simulation tool for multi-physical modelling and is intended for modelling more complex engineering problems.

2. The cantilever beam test conducted with linear parallel bond model have proved that the performance of DEM in dealing with structural deformation and failure can acceptably match numerical and analytical results.
3. The developed integrated model is applied to simulate 3D printing of fibre reinforced composites and injection moulding of short fibre composites to investigate the special distribution of composites. The simulation results successfully showed the moving tendency of composites.

10.2 Future works

The current model is not the ultimate version as it still has some deficiencies such as:

1. 2D simulation is not able to completely simplify complex engineering problems,
2. Large-scale modelling is limited by the number of particle and calculation algorithm,
3. Only fluid dynamics is incorporated into current model, which is far behind the ultimate goal in solving real engineering problems.

Therefore, some potential improvements can be made in the future. The coupled model will be expanded from 2D to 3D simulation with GPU acceleration in order to become a practical tool for predicting and optimizing engineering problems. The simulation of fluid-structure interaction with fracture will be validated through laboratory experiment and further improvement will be made by considering heat transfer, consolidation, and crystallisation, together with comprehensive validations against experiments.

List of References

- [1] Bungartz H-J, Schäfer M. Fluid-structure interaction: modelling, simulation, optimisation: Springer Science & Business Media; 2006.
- [2] Mucha PJ, Tee S-Y, Weitz DA, Shraiman BI, Brenner MP. A model for velocity fluctuations in sedimentation. *Journal of Fluid Mechanics*. 2004;501:71-104.
- [3] Tornberg A-K, Shelley MJ. Simulating the dynamics and interactions of flexible fibers in Stokes flows. *Journal of Computational Physics*. 2004;196:8-40.
- [4] Wang J, Layton A. Numerical simulations of fiber sedimentation in Navier-Stokes flows. *Communications in Computational Physics*. 2009;5:61.
- [5] Liu WK, Liu Y, Farrell D, Zhang L, Wang XS, Fukui Y, et al. Immersed finite element method and its applications to biological systems. *Computer methods in applied mechanics and engineering*. 2006;195:1722-49.
- [6] Haase W. Unsteady aerodynamics including fluid/structure interaction. *Air & Space Europe*. 2001;3:83-6.
- [7] Zhang W, Jiang Y, Ye Z. Two better loosely coupled solution algorithms of CFD based aeroelastic simulation. *Engineering Applications of Computational Fluid Mechanics*. 2007;1:253-62.
- [8] Kalitzin G, Iaccarino G. Toward immersed boundary simulation of high Reynolds number flows. STANFORD UNIV CA; 2003.

- [9] Yang J, Balaras E. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*. 2006;215:12-40.
- [10] Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*. 2000;161:35-60.
- [11] Udaykumar H, Shyy W, Rao M. ELAFINT-A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries. p. 1996.
- [12] Udaykumar HS, Mittal R, Rampunggoon P, Khanna A. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*. 2001;174:345-80.
- [13] Hoburg JF, Melcher JR. Internal electrohydrodynamic instability and mixing of fluids with orthogonal field and conductivity gradients. *Journal of Fluid Mechanics*. 1976;73:333-51.
- [14] Grigoriadis DGE, Kassinos SC, Votyakov EV. Immersed boundary method for the MHD flows of liquid metals. *Journal of Computational Physics*. 2009;228:903-20.
- [15] Nobili M, Morbiducci U, Ponzini R, Del Gaudio C, Balducci A, Grigioni M, et al. Numerical simulation of the dynamics of a bileaflet prosthetic heart valve using a fluid–structure interaction approach. *Journal of biomechanics*. 2008;41:2539-50.
- [16] Liu Y, Liu WK. Rheology of red blood cell aggregation by computer simulation. *Journal of Computational Physics*. 2006;220:139-54.

- [17] Hou G, Wang J, Layton A. Numerical methods for fluid-structure interaction—a review. *Communications in Computational Physics*. 2012;12:337-77.
- [18] Strang G, Fix GJ. *An analysis of the finite element method*: Prentice-hall Englewood Cliffs, NJ; 1973.
- [19] Nie X, Doolen GD, Chen S. Lattice-Boltzmann simulations of fluid flows in MEMS. *Journal of Statistical Physics*. 2002;107:279-89.
- [20] Colagrossi A, Landrini M. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*. 2003;191:448-75.
- [21] Langston PA, Tüzün U, Heyes DM. Discrete element simulation of granular flow in 2D and 3D hoppers: dependence of discharge rate and wall stress on particle interactions. *Chemical Engineering Science*. 1995;50:967-87.
- [22] Tan Y, Yang D, Sheng Y. Discrete element method (DEM) modeling of fracture and damage in the machining process of polycrystalline SiC. *Journal of the European ceramic society*. 2009;29:1029-37.
- [23] Antoci C, Gallati M, Sibilla S. Numerical simulation of fluid–structure interaction by SPH. *Computers & Structures*. 2007;85:879-90.
- [24] Ren B, Jin Z, Gao R, Wang Y-x, Xu Z-l. SPH-DEM Modeling of the Hydraulic Stability of 2D Blocks on a Slope. *Journal of Waterway, Port, Coastal, and Ocean Engineering*. 2013;140:04014022.
- [25] Rabczuk T, Gracie R, Song J-H, Belytschko T. Immersed particle method for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*. 2010;22:48.

- [26] Rabczuk T, Belytschko T. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*. 2007;196:2777-99.
- [27] Han K, Feng YT, Owen DRJ. Numerical simulations of irregular particle transport in turbulent flows using coupled LBM-DEM. *Computer Modeling in Engineering and Sciences*. 2007;18:87.
- [28] Hirsch C. *The Basic Equations of Fluid Dynamics. Numerical computation of internal and external flows*. 1988;1:8-23.
- [29] Zienkiewicz OC, Taylor RL. *The finite element method: solid mechanics*: Butterworth-heinemann; 2000.
- [30] Liu G-R, Quek SS. *The finite element method: a practical course*: Butterworth-Heinemann; 2013.
- [31] Cheng JH, Kikuchi N. A mesh re-zoning technique for finite element simulations of metal forming processes. *International Journal for Numerical Methods in Engineering*. 1986;23:219-28.
- [32] Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics*. 1982;46:342-68.
- [33] Masud A, Bhanabagvanwala M, Khurram RA. An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction. *Computers & fluids*. 2007;36:77-91.
- [34] Mair HU. Hydrocodes for structural response to underwater explosions. *Shock and Vibration*. 1999;6:81-96.
- [35] Benson DJ. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer methods in applied mechanics and engineering*. 1992;99:235-394.

- [36] Belytschko T, Liu WK, Moran B, Elkhodary K. Nonlinear finite elements for continua and structures: John Wiley & sons; 2013.
- [37] Liu GR, Lam KY, Lu C. Computational simulation of sympathetic explosion in a high performance magazine. p. 65-75.
- [38] Donea J, Giuliani S, Halleux J-P. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. Computer methods in applied mechanics and engineering. 1982;33:689-723.
- [39] Donea J, Fasoli-Stella P, Giuliani S. Lagrangian and Eulerian finite element techniques for transient fluid-structure interaction problems. Structural mechanics in reactor technology 1977.
- [40] Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. Monthly notices of the royal astronomical society. 1977;181:375-89.
- [41] Monaghan JJ. Smoothed particle hydrodynamics. Annual review of astronomy and astrophysics. 1992;30:543-74.
- [42] Liu MB, Liu GR. Smoothed particle hydrodynamics (SPH): an overview and recent developments. Archives of computational methods in engineering. 2010;17:25-76.
- [43] Morris JP. Simulating surface tension with smoothed particle hydrodynamics. International journal for numerical methods in fluids. 2000;33:333-53.
- [44] Libersky LD, Petschek AG. Smooth particle hydrodynamics with strength of materials. Advances in the free-Lagrange method including contributions on adaptive gridding and the smooth particle hydrodynamics method: Springer; 1991. p. 248-57.

- [45] Monaghan JJ. SPH without a tensile instability. *Journal of Computational Physics*. 2000;159:290-311.
- [46] Chen JK, Beraun JE, Jih CJ. An improvement for tensile instability in smoothed particle hydrodynamics. *Computational Mechanics*. 1999;23:279-87.
- [47] Liu MB, Liu GR. Restoring particle consistency in smoothed particle hydrodynamics. *Applied numerical mathematics*. 2006;56:19-36.
- [48] Koshizuka S, Oka Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering*. 1996;123:421-34.
- [49] Koshizuka S, Nobe A, Oka Y. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International journal for numerical methods in fluids*. 1998;26:751-69.
- [50] Ataie-Ashtiani B, Farhadi L. A stable moving-particle semi-implicit method for free surface flows. *Fluid dynamics research*. 2006;38:241-56.
- [51] Khayyer A, Gotoh H. Modified moving particle semi-implicit methods for the prediction of 2D wave impact pressure. *Coastal Engineering*. 2009;56:419-40.
- [52] Cundall PA. The measurement and analysis of accelerations in rock slopes. PhD Thesis, Imperial College of Science & Technology. 1971.
- [53] Jiang M, Yu HS, Leroueil S. A simple and efficient approach to capturing bonding effect in naturally microstructured sands by discrete element method. *International Journal for Numerical Methods in Engineering*. 2007;69:1158-93.
- [54] Lucy LB. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*. 1977;82:1013-24.

- [55] Gingold RA, Monaghan JJ. Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*. 1982;46:429-53.
- [56] Hu XY, Adams NA. Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows. *Physics of Fluids*. 2006;18:101702.
- [57] Swegle JW, Hicks DL, Attaway SW. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*. 1995;116:123-34.
- [58] Johnson GR. Artificial viscosity effects for SPH impact computations. *International Journal of Impact Engineering*. 1996;18:477-88.
- [59] Randles PW, Libersky LD. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer methods in applied mechanics and engineering*. 1996;139:375-408.
- [60] Chen JK, Beraun JE. A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. *Computer methods in applied mechanics and engineering*. 2000;190:225-39.
- [61] Batra RC, Zhang GM. Analysis of adiabatic shear bands in elasto-thermo-viscoplastic materials by modified smoothed-particle hydrodynamics (MSPH) method. *Journal of Computational Physics*. 2004;201:172-90.
- [62] Dyka CT, Randles PW, Ingel RP. Stress points for tension instability in SPH. *International Journal for Numerical Methods in Engineering*. 1997;40:2325-41.
- [63] Dilts GA. Moving least-squares particle hydrodynamics II: conservation and boundaries. *International Journal for Numerical Methods in Engineering*. 2000;48:1503-24.

- [64] Bonet J, Kulasegaram S. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*. 2000;47:1189-214.
- [65] Chen J-S, Pan C, Wu C-T, Liu WK. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer methods in applied mechanics and engineering*. 1996;139:195-227.
- [66] Hu XY, Adams NA. A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics*. 2006;213:844-61.
- [67] Monaghan JJ, Kocharyan A. SPH simulation of multi-phase flow. *Computer Physics Communications*. 1995;87:225-35.
- [68] De Lefte M, Le Touzé D, Alessandrini B. SPH modeling of shallow-water coastal flows. *Journal of Hydraulic Research*. 2010;48:118-25.
- [69] Dalrymple RA, Knio O. SPH modelling of water waves. p. 779-87.
- [70] Tanaka N, Takano T. Microscopic-scale simulation of blood flow using SPH method. *International Journal of Computational Methods*. 2005;2:555-68.
- [71] Cleary PW. Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling*. 1998;22:981-93.
- [72] Rook R, Yildiz M, Dost S. Modeling transient heat transfer using SPH and implicit time integration. *Numerical Heat Transfer, Part B: Fundamentals*. 2007;51:1-23.
- [73] Shen HT, Su J, Liu L. SPH simulation of river ice dynamics. *Journal of Computational Physics*. 2000;165:752-70.

- [74] Gutfraind R, Savage SB. Smoothed particle hydrodynamics for the simulation of broken-ice fields: Mohr–Coulomb-type rheology and frictional boundary conditions. *Journal of Computational Physics*. 1997;134:203-15.
- [75] Cleary PW, Prakash M, Ha J. Novel applications of smoothed particle hydrodynamics (SPH) in metal forming. *Journal of materials processing technology*. 2006;177:41-8.
- [76] Cundall PA. A computer model for simulating progressive, large scale movement in blocky rock systems. p. 129-36.
- [77] Dobry R, Tang-Tat Ng AND. Discrete modelling of stress-strain behaviour of granular media at small and large strains. *Engineering Computations*. 1992;9:129-43.
- [78] Boon CW, Houlsby GT, Utili S. A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method. *Computers and Geotechnics*. 2012;44:73-82.
- [79] Peters JF, Hopkins MA, Kala R, Wahl RE. A poly-ellipsoid particle for non-spherical discrete element method. *Engineering Computations*. 2009;26:645-57.
- [80] Ting JM. A robust algorithm for ellipse-based discrete element modelling of granular materials. *Computers and Geotechnics*. 1992;13:175-86.
- [81] Favier JF, Abbaspour-Fard MH, Kremmer M, Raji AO. Shape representation of axi-symmetrical, non-spherical particles in discrete element simulation using multi-element model particles. *Engineering Computations*. 1999;16:467-80.

- [82] Feng YT, Owen DRJ. An augmented spatial digital tree algorithm for contact detection in computational mechanics. *International Journal for Numerical Methods in Engineering*. 2002;55:159-76.
- [83] Mio H, Shimosaka A, Shirakawa Y, Hidaka J. Cell optimization for fast contact detection in the discrete element method algorithm. *Advanced Powder Technology*. 2007;18:441-53.
- [84] Munjiza A, Andrews KRF. NBS contact detection algorithm for bodies of similar size. *International Journal for Numerical Methods in Engineering*. 1998;43:131-49.
- [85] Williams JR, Perkins E, Cook B. A contact algorithm for partitioning N arbitrary sized objects. *Engineering Computations*. 2004;21:235-48.
- [86] Liu Q, Gu Y, Zhuang P, Liu F, Nie YF. An implicit RBF meshless approach for time fractional diffusion equations. *Computational Mechanics*. 2011;48:1-12.
- [87] Soares Jr D, Sladek J, Sladek V. Dynamic analysis by meshless local Petrov-Galerkin formulations considering a time-marching scheme based on implicit Green's functions. *Computer Modeling in Engineering and Sciences (CMES)*. 2009;19:115.
- [88] Farhat C, Lesoinne M, Maman N. Mixed explicit/implicit time integration of coupled aeroelastic problems: Three-field formulation, geometric conservation and distributed solution. *International journal for numerical methods in fluids*. 1995;21:807-35.
- [89] O'Sullivan C, Bray JD. Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme. *Engineering Computations*. 2004;21:278-303.

- [90] Ketterhagen WR, am Ende MT, Hancock BC. Process modeling in the pharmaceutical industry using the discrete element method. *Journal of pharmaceutical sciences*. 2009;98:442-70.
- [91] Fu X, Dutt M, Bentham AC, Hancock BC, Cameron RE, Elliott JA. Investigation of particle packing in model pharmaceutical powders using X-ray microtomography and discrete element method. *Powder Technology*. 2006;167:134-40.
- [92] Su O, Akcin NA. Numerical simulation of rock cutting using the discrete element method. *International Journal of Rock Mechanics and Mining Sciences*. 2011;48:434-42.
- [93] Scholtès LUC, Donzé F-V. Modelling progressive failure in fractured rock masses using a 3D discrete element method. *International Journal of Rock Mechanics and Mining Sciences*. 2012;52:18-30.
- [94] Martin CL, Bouvard D, Shima S. Study of particle rearrangement during powder compaction by the discrete element method. *Journal of the Mechanics and Physics of Solids*. 2003;51:667-93.
- [95] Skrinjar O, Larsson P-L. On discrete element modelling of compaction of powders with size ratio. *Computational Materials Science*. 2004;31:131-46.
- [96] Lu M, McDowell GR. The importance of modelling ballast particle shape in the discrete element method. *Granular matter*. 2007;9:69.
- [97] Saussine G, Cholet C, Gautier PE, Dubois F, Bohatier C, Moreau J-J. Modelling ballast behaviour under dynamic loading. Part 1: A 2D polygonal discrete element method approach. *Computer methods in applied mechanics and engineering*. 2006;195:2841-59.

- [98] Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *geotechnique*. 1979;29:47-65.
- [99] Dunne T. An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation. *International journal for numerical methods in fluids*. 2006;51:1017-39.
- [100] Takagi S, Sugiyama K, li S, Matsumoto Y. A review of full Eulerian methods for fluid structure interaction problems. *Journal of Applied Mechanics*. 2012;79:010911.
- [101] Ahmed S, Leithner R, Kosyna G, Wulff D. Increasing reliability using FEM–CFD. *World Pumps*. 2009;2009:35-9.
- [102] Kim S-H, Choi J-B, Park J-S, Choi Y-H, Lee J-H. A coupled CFD-FEM analysis on the safety injection piping subjected to thermal stratification. *Nuclear Engineering and Technology*. 2013;45:237-48.
- [103] Peksen M. 3D transient multiphysics modelling of a complete high temperature fuel cell system using coupled CFD and FEM. *International Journal of Hydrogen Energy*. 2014;39:5137-47.
- [104] Peksen M, Peters R, Blum L, Stolten D. 3D coupled CFD/FEM modelling and experimental validation of a planar type air pre-heater used in SOFC technology. *International Journal of Hydrogen Energy*. 2011;36:6851-61.
- [105] Hou G, Wang J, Layton A. Numerical methods for fluid-structure interaction—a review. *Commun Comput Phys*. 2012;12:337-77.
- [106] Blocken B, Stathopoulos T, Carmeliet J. CFD simulation of the atmospheric boundary layer: wall function problems. *Atmospheric Environment*. 2007;41:238-52.

- [107] Lee N-S, Bathe K-J. Error indicators and adaptive remeshing in large deformation finite element analysis. *Finite Elements in Analysis and Design*. 1994;16:99-139.
- [108] Liu G-R, Liu MB. *Smoothed particle hydrodynamics: a meshfree particle method*: World Scientific; 2003.
- [109] De Rosis A, Falcucci G, Ubertini S, Ubertini F. A coupled lattice Boltzmann-finite element approach for two-dimensional fluid–structure interaction. *Computers & Fluids*. 2013;86:558-68.
- [110] Haddad H, Guessasma M, Fortin J. Heat transfer by conduction using DEM–FEM coupling method. *Computational Materials Science*. 2014;81:339-47.
- [111] Canelas RB, Crespo AJC, Domínguez JM, Ferreira RML, Gómez-Gesteira M. SPH–DCDEM model for arbitrary geometries in free surface solid–fluid flows. *Computer Physics Communications*. 2016;202:131-40.
- [112] Robinson M, Ramaioli M, Luding S. Fluid–particle flow simulations using two-way-coupled mesoscale SPH–DEM and validation. *International journal of multiphase flow*. 2014;59:121-34.
- [113] Birnbaum NK, Francis NJ, Gerber BI. Coupled techniques for the simulation of fluid-structure and impact problems. *Computer Assisted Mechanics and Engineering Sciences*. 1999;6:295-311.
- [114] Rafiee A, Thiagarajan KP. An SPH projection method for simulating fluid-hypoelastic structure interaction. *Computer Methods in Applied Mechanics and Engineering*. 2009;198:2785-95.
- [115] Fries TP. A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*. 2008;75:503-32.

- [116] Zi G, Belytschko T. New crack-tip elements for XFEM and applications to cohesive cracks. *International Journal for Numerical Methods in Engineering*. 2003;57:2221-40.
- [117] Vu-Bac N, Nguyen-Xuan H, Chen L, Lee C-K, Zi G, Zhuang X, et al. A phantom-node method with edge-based strain smoothing for linear elastic fracture mechanics. *Journal of Applied Mathematics*. 2013;2013.
- [118] Cai Y, Zhu H, Zhuang X. A continuous/discontinuous deformation analysis (CDDA) method based on deformable blocks for fracture modeling. *Frontiers of Structural and Civil Engineering*. 2013;7:369-78.
- [119] Zhuang X, Augarde CE, Mathisen KM. Fracture modeling using meshless methods and level sets in 3D: framework and modeling. *International Journal for Numerical Methods in Engineering*. 2012;92:969-98.
- [120] Rabczuk T, Zi G. A meshfree method based on the local partition of unity for cohesive cracks. *Computational Mechanics*. 2007;39:743-60.
- [121] Rabczuk T, Areias PMA, Belytschko T. A meshfree thin shell method for non-linear dynamic fracture. *International Journal for Numerical Methods in Engineering*. 2007;72:524-48.
- [122] Tan Y, Zhang H, Yang D, Jiang S, Song J, Sheng Y. Numerical simulation of concrete pumping process and investigation of wear mechanism of the piping wall. *Tribology International*. 2012;46:137-44.
- [123] Yang D, Sheng Y, Ye J, Tan Y. Discrete element modeling of the microbond test of fiber reinforced composite. *Computational materials science*. 2010;49:253-9.

- [124] Tavares FA, Plesha ME. Discrete element method for modelling solid and particulate materials. *International Journal for Numerical Methods in Engineering*. 2007;70:379-404.
- [125] Chen JK, Beraun JE, Carney TC. A corrective smoothed particle method for boundary value problems in heat conduction. *International Journal for Numerical Methods in Engineering*. 1999;46:231-52.
- [126] Monaghan JJ, Lattanzio JC. A refined particle method for astrophysical problems. *Astronomy and Astrophysics*. 1985;149:135-43.
- [127] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*. 1995;4:389-96.
- [128] Monaghan JJ. Smoothed particle hydrodynamics. *Reports on progress in physics*. 2005;68:1703.
- [129] Morris JP, Fox PJ, Zhu Y. Modeling low Reynolds number incompressible flows using SPH. *Journal of computational physics*. 1997;136:214-26.
- [130] Morris JP, Monaghan JJ. A switch to reduce SPH viscosity. *Journal of Computational Physics*. 1997;136:41-50.
- [131] Randles PW, Libersky LD. Normalized SPH with stress points. *International Journal for Numerical Methods in Engineering*. 2000;48:1445-62.
- [132] Monaghan JJ. Simulating free surface flows with SPH. *Journal of computational physics*. 1994;110:399-406.
- [133] Mindlin RD, Deresiewica H. Elastic spheres in contact under varying oblique forces. *Journal of Applied Mechanics*. 2014;20.
- [134] Itasca Consulting Group I. PFC 5.0 documentation. 2011.

- [135] Potyondy DO, Cundall PA. A bonded-particle model for rock. *International Journal of Rock Mechanics and Mining Sciences*. 2004;41:1329-64.
- [136] Yang D, Sheng Y, Ye J, Tan Y. Dynamic simulation of crack initiation and propagation in cross-ply laminates by DEM. *Composites Science and Technology*. 2011;71:1410-8.
- [137] Anderson TB, Jackson R. Fluid mechanical description of fluidized beds. Equations of motion. *Industrial & Engineering Chemistry Fundamentals*. 1967;6:527-39.
- [138] Sun X, Sakai M, Yamada Y. Three-dimensional simulation of a solid–liquid flow by the DEM–SPH method. *Journal of Computational Physics*. 2013;248:147-76.
- [139] Ergun S. Fluid flow through packed columns. *Chem Eng Prog*. 1952;48:89-94.
- [140] Wen CY, Yu Y. *Mechanics of fluidization*. 62 ed. p. 100.
- [141] Crowe CT, Schwarzkopf JD, Sommerfeld M, Tsuji Y. *Multiphase flows with droplets and particles*: CRC press; 2011.
- [142] Jánosi IM, Jan D, Szabó KG, Tél T. Turbulent drag reduction in dam-break flows. *Experiments in Fluids*. 2004;37:219-29.
- [143] Koshizuka S, Oka Y, Tamako H. A particle method for calculating splashing of incompressible viscous fluid. American Nuclear Society, Inc., La Grange Park, IL (United States); 1995.
- [144] Zhang S, Kuwabara S, Suzuki T, Kawano Y, Morita K, Fukuda K. Simulation of solid–fluid mixture flow using moving particle methods. *Journal of Computational Physics*. 2009;228:2552-65.

- [145] McGuire W, Gallagher RH, Ziemian RD. Matrix structural analysis2000.
- [146] Wu J, Shu C. An improved immersed boundary-lattice Boltzmann method for simulating three-dimensional incompressible flows. *Journal of Computational Physics*. 2010;229:5022-42.
- [147] Zhang H, Tan Y, Shu S, Niu X, Trias FX, Yang D, et al. Numerical investigation on the role of discrete element method in combined LBM–IBM–DEM modeling. *Computers & Fluids*. 2014;94:37-48.
- [148] Nash NH, Young TM, McGrail PT, Stanley WF. Inclusion of a thermoplastic phase to improve impact and post-impact performances of carbon fibre reinforced thermosetting composites—a review. *Materials & Design*. 2015;85:582-97.
- [149] Wang X, Jiang M, Zhou Z, Gou J, Hui D. 3D printing of polymer matrix composites: a review and prospective. *Composites Part B: Engineering*. 2017;110:442-58.
- [150] Ning F, Cong W, Qiu J, Wei J, Wang S. Additive manufacturing of carbon fiber reinforced thermoplastic composites using fused deposition modeling. *Composites Part B: Engineering*. 2015;80:369-78.
- [151] Compton BG, Lewis JA. 3D-printing of lightweight cellular composites. *Advanced materials*. 2014;26:5930-5.
- [152] Markforged I. THE MARK TWO. 2017.
- [153] envisionTEC. Large Format 3D Printer For Industrial Composites. 2017.
- [154] 3DPRINT.COM. Robot Bike Company Blends 3D Printed Titanium with Carbon Fiber for R160 Mountain Bike Frame. 2017.

- [155] Llewellyn-Jones TM, Drinkwater BW, Trask RS. 3D printed components with ultrasonically arranged microscale structure. *Smart Materials and Structures*. 2016;25:02LT1.
- [156] Martin JJ, Fiore BE, Erb RM. Designing bioinspired composite reinforcement architectures via 3D magnetic printing. *Nature communications*. 2015;6.
- [157] Spackman CC, Frank CR, Picha KC, Samuel J. 3D printing of fiber-reinforced soft composites: Process study and material characterization. *Journal of Manufacturing Processes*. 2016;23:296-305.
- [158] Ning F, Cong W, Hu Z, Huang K. Additive manufacturing of thermoplastic matrix composites using fused deposition modeling: A comparison of two reinforcements. *Journal of Composite Materials*. 2017:0021998317692659.
- [159] Van Der Klift F, Koga Y, Todoroki A, Ueda M, Hirano Y, Matsuzaki R. 3D printing of continuous carbon fibre reinforced thermo-plastic (CFRTP) tensile test specimens. *Open Journal of Composite Materials*. 2015;6:18.
- [160] Wu K, Yang D, Wright N. A coupled SPH-DEM model for fluid-structure interaction problems with free-surface flow and structural failure. *Computers & Structures*. 2016;177:141-61.
- [161] Wu K, Wan L, Yang D. An integrated particle model for fluid-particle-structure interaction problems with free-surface flow and structural failure. *Journal of Fluids and Structures*. 2017.
- [162] Makino M, Fukuzawa D, Murashima T, Kawakami M, Furukawa H. Analysis of deposition modeling by particle method simulation. *Microsystem Technologies*. 2017;23:1177-81.

- [163] Steuben JC, Iliopoulos AP, Michopoulos JG. Discrete element modeling of particle-based additive manufacturing processes. *Computer Methods in Applied Mechanics and Engineering*. 2016;305:537-61.
- [164] Yashiro S, Sasaki H, Sakaida Y. Particle simulation for predicting fiber motion in injection molding of short-fiber-reinforced composites. *Composites Part A: Applied Science and Manufacturing*. 2012;43:1754-64.
- [165] Yashiro S, Okabe T, Matsushima K. A numerical approach for injection molding of short-fiber-reinforced plastics using a particle method. *Advanced Composite Materials*. 2011;20:503-17.
- [166] Advani SG, Tucker lii CL. The use of tensors to describe and predict fiber orientation in short fiber composites. *Journal of rheology*. 1987;31:751-84.
- [167] Gupta M, Wang K. Fiber orientation and mechanical properties of short-fiber-reinforced injection-molded composites: Simulated and experimental results. *Polymer Composites*. 1993;14:367-82.
- [168] Chung S, Kwon T. Coupled analysis of injection molding filling and fiber orientation, including in-plane velocity gradient effect. *Polymer Composites*. 1996;17:859-72.
- [169] Ramazani A, Ait-Kadi A, Grmela M. Rheological modelling of short fiber thermoplastic composites. *Journal of Non-Newtonian Fluid Mechanics*. 1997;73:241-60.
- [170] Sun X, Gan Y, Lasecki J, Zeng D, Qi L, Li L, et al. Evaluation of Fiber Orientation Prediction of Moldflow Using an Injection Molded IP Panel. *Proceedings of International Manufacturing Science and Engineering Conference*. Wisconsin, USA2013.

- [171] Oumer AN, Hamidi NM, Sahat IM. Numerical prediction of flow induced fibers orientation in injection molded polymer composites. The 3rd International Conference on Mechanical Engineering Research Pahang, Malaysia2015. p. 012066.
- [172] Modhaffar I, Gueraoui K, El-tourroug H, Men-la-yakhaf S. Numerical study of short fiber orientation in simple injection molding processes. Proceedings of AIP Conference Proceedings. Fethiye, Turkey2014.
- [173] Thi TBN, Morioka M, Yokoyama A, Hamanaka S, Yamashita K, Nonomura C. Numerical prediction of fiber orientation in injection-molded short-fiber/thermoplastic composite parts with experimental validation. Proceedings of AIP Conference Proceedings. Fethiye, Turkey2014.
- [174] Folgar F, Charles L. Tucker I. Orientation Behavior of Fibers in Concentrated Suspensions. Journal of Reinforced Plastics and Composites. 1984;3:98-119.
- [175] Yamamoto S, Matsuoka T. Dynamic simulation of microstructure and rheology of fiber suspensions. Polymer Engineering & Science. 1996;36:2396-403.
- [176] Yamamoto S, Matsuoka T. Dynamic simulation of rod-like and plate-like particle dispersed systems. Computational Materials Science. 1999;14:169-76.
- [177] Koshizuka S. Current achievements and future perspectives on particle simulation technologies for fluid dynamics and heat transfer. Journal of nuclear science and technology. 2011;48:155-68.
- [178] Von Turkovich R, Erwin L. Fiber fracture in reinforced thermoplastic processing. Polymer Engineering & Science. 1983;23:743-9.

- [179] Wu K, Yang D, Wright N, Khan A. An integrated particle model for fluid-particle-structure interaction problems with free-surface flow and structural failure. *Journal of fluids and structures*. 2017.
- [180] Jeffery GB. The motion of ellipsoidal particles immersed in a viscous fluid. *Proceedings of Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences: The Royal Society*; 1922.

Appendix 1: Publication list

1. Wu K, Yang D, Wright N. A coupled SPH-DEM model for fluid-structure interaction problems with free-surface flow and structural failure. *Computers & Structures*. 2016;177:141-61.
2. Yang D, Wu K, Wan L, Sheng Y. "A Particle Element Approach for Modelling the 3D Printing Process of Fibre Reinforced Polymer Composites." *Journal of Manufacturing and Materials Processing* 1.1 (2017): 10.
3. Wu K, Yang D, Wright N, M. A. I. Khan. A unified particle model for fluid-particle-structure interaction problems with free-surface flow and structural failure.
4. Wu K, Wan L, Yang D. Two-dimensional simulation of the injection molding process of short-fiber composite by the coupling of Smoothed Particle Hydrodynamics and Discrete Element Method. (paper recently submitted to the journal "Journal of Manufacturing Processes")

Appendix 2: The main structure of SPH code in C++

The following is the main structure of SPH code written in C++. Basically this code is based on the loop of searching each particle followed by a series of SPH implementation for fluid-solid interaction problems:

```
case FPSIFunction:      {
                        double dp = 0.003;
                        double eps = 0.2*dp;
//coefficent for position correction
                        double h = 1.25*dp;
                        double xmax = 0.201;
                        double xmin = -0.0015;
                        double ymax = 1.78;
                        double viscous = 8.9e-
4;
                        double height = 0.1;
                        double rho0 = 1000.0;
//contact model for
dem/dem and dem/wall
                        double kn = 5.3e5;
                        double ks = 0.25*kn;
                        double fric = 0.2;
                        double dp_nratio =
0.1;
                        double dist;
                        double kc =
7.0/(64.0*M_PI*pow(h,2));
                        double w;
                        double dwx;
                        double dwy;
                        double mass =
                        double vmax =
rho0*pow(dp,2);
                        double cs = 10.0*vmax;
                        double k =
sqrt(2.0*9.81*height);
                        double rho;
                        double initialw =
pow(cs,2)*rho0/7.0;
                        double rho;
                        double initialw =
kc*(pow(2.0-dp/h,4)*(1.0+2.0*dp/h));
                        IModuleBall *m = prog-
>findInterface<IModuleBall>());
                        if (!m) throw Exception("The clump module was
not found.");
                        IGlobals *g = prog-
>findInterface<IGlobals>());
                        if (!m) throw
Exception("The clump module was not found.");
```

```
>getTimestep();

double dt = g-

double dens;

for (IIterator<IBall>

it(m->getIContainerBall());!it.atEnd();++it) {

    IBall *ball = it;

    double vfraction

    double

    double pressure;

    if (ball-

    {

        dens =

        pressure

    = 1.0;

    demradius;

    >getIThing()->getGroupName()!="sph" && ball->getIThing()-

    >getGroupName()!="particle")

    {

        dens =

        pressure

    = 0.0;

    IFishParam *ex1setptr = ball->getIThing()->setExtra(1);

    double ex1 =

    dens;

    *ex1setptr = ex1;

    IFishParam

    *ex2setptr = ball->getIThing()->setExtra(2);

    double ex2 =

    pressure;

    *ex2setptr = ex2;

    }

    >getIThing()->getGroupName()=="sph")

    {

        FArray<IContact *> list;

        ball->getIPiece()-

    >getContactList(&list);

    for

    (quintptr i=0; i<list.size(); ++i)

    {

        *c = list.at(i);

        IContact

        if

        (c->getEnd2()->getIThing()->getTypeName() != "ball" || c->getEnd1()-

        >getIThing()->getTypeName() != "ball") continue;

        if

        (c->getEnd2()->getIThing()->getGroupName() == "particle" || c->getEnd1()-

        >getIThing()->getGroupName() == "particle")

        {

            DVect pos1(0.0);

            DVect

            pos2(0.0);
```



```
quintptr ID1 = c->getEnd1()->getIThing()->getID();
quintptr ID2 = c->getEnd2()->getIThing()->getID();
quintptr ID3 = ball->getIPiece()->getIThing()->getID();

    if (ID3==ID1)
    {
        IBall *ball2 = m->findBallWithID(ID2);

        demradius = ball2->getRadius();

pos1 = ball->getIPieceMechanical()->getPosition();
pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

if (q<=2.0)
{
    w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
}
else
{
    w =0.0;
}
}

    else if (ID3==ID2)
    {
IBall *ball2 = m->findBallWithID(ID1);

        demradius = ball2->getRadius();

pos1 = ball->getIPieceMechanical()->getPosition();

pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));
```

```
double q = dist / h;

if (q<=2.0)
{
    w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
}
else
{
    w =0.0;
}

}
vfraction -= w*pow(demradius,2);
}
}
dens =
pressure

ball->getIBodyMechanical()->getDensity();
= k*(pow(dens/rho0,7)-1.0);

    IFishParam *ex1setptr = ball->getIThing()->setExtra(1);
double ex1 =
dens*vfraction;
    *ex1setptr = ex1;
IFishParam
*ex2setptr = ball->getIThing()->setExtra(2);
double ex2 =
pressure;
    *ex2setptr = ex2;
IFishParam *ex3setptr = ball->getIThing()->setExtra(3);
double ex3 =
vfraction;
    *ex3setptr = ex3;
}
}
for (IIterator<IBall>
it(m->getIContainerBall());!it.atEnd();++it) {
    IBall *ball =
double
double
double numetrho
= 0.0;
```



```
c->setProperty(dp_nratio, "dp_nratio");
}

else if (c->getEnd2()->getIThing()->getGroupName() == "particle" && c-
>getEnd1()->getIThing()->getGroupName() == "particle")
{

c->setModelName("linear");
c->setProperty(kn, "kn");

c->setProperty(ks, "ks");

c->setProperty(fric, "fric");

c->setProperty(dp_nratio, "dp_nratio");

DVect pos1(0.0);
DVect
pos2(0.0);

quintptr ID1 = c->getEnd1()->getIThing()->getID();
quintptr ID2 = c->getEnd2()->getIThing()->getID();
quintptr ID3 = ball->getIPiece()->getIThing()->getID();

if (ID3==ID1)
{
IBall *ball2 = m->findBallWithID(ID2);

pos1 = ball->getIPieceMechanical()->getPosition();
pos2 = ball2->getIPieceMechanical()->getPosition();

DVect vel1 = ball->getVelocity();
DVect vel2 = ball2->getVelocity();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double demradius = ball->getRadius();
double cutoff = 2.0*demradius;

if (dist >= cutoff) continue;

lubrixindi = -3.0*M_PI*viscous*pow(cutoff,2)/8.0/(dist-
cutoff)*((vel1.x()-vel2.x())*(pos1.x()-pos2.x())+(vel1.y()-
vel2.y()*(pos1.y()-pos2.y()))/((pos1.x()-pos2.x()*(pos1.x()-
pos2.x())+(pos1.y()-pos2.y()*(pos1.y()-pos2.y()))*(pos1.x()-pos2.x()));

lubriyindi = -3.0*M_PI*viscous*pow(cutoff,2)/8.0/(dist-
cutoff)*((vel1.x()-vel2.x()*(pos1.x()-pos2.x())+(vel1.y()-
```

```
vel2.y()*(pos1.y()-pos2.y()))/((pos1.x()-pos2.x()*(pos1.x()-
pos2.x())+(pos1.y()-pos2.y()*(pos1.y()-pos2.y()*(pos1.y()-pos2.y()));

    //stream<<"lubrixindi is "<<lubrixindi<<endl;

    //stream<<"lubriyindi is "<<lubriyindi<<endl;

}

else if (ID3==ID2)

{

    IBall *ball2 = m->findBallWithID(ID1);

    pos1 = ball->getIPieceMechanical()->getPosition();
pos2 = ball2->getIPieceMechanical()->getPosition();

DVect vel1 = ball->getVelocity();

    DVect vel2 = ball2->getVelocity();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

    double demradius = ball->getRadius();

    double cutoff = 2.0*demradius;

    if (dist >= cutoff) continue;

    lubrixindi = -3.0*M_PI*viscous*pow(cutoff,2)/8.0/(dist-
cutoff)*((vel1.x()-vel2.x()*(pos1.x()-pos2.x())+(vel1.y()-
vel2.y()*(pos1.y()-pos2.y())/((pos1.x()-pos2.x()*(pos1.x()-
pos2.x())+(pos1.y()-pos2.y()*(pos1.y()-pos2.y()*(pos1.x()-pos2.x()));

    lubriyindi = -3.0*M_PI*viscous*pow(cutoff,2)/8.0/(dist-
cutoff)*((vel1.x()-vel2.x()*(pos1.x()-pos2.x())+(vel1.y()-
vel2.y()*(pos1.y()-pos2.y())/((pos1.x()-pos2.x()*(pos1.x()-
pos2.x())+(pos1.y()-pos2.y()*(pos1.y()-pos2.y()*(pos1.y()-pos2.y()));

    //stream<<"lubrixindi is "<<lubrixindi<<endl;

    //stream<<"lubriyindi is "<<lubriyindi<<endl;

}

lubritotalx += lubrixindi;

lubritotaly += lubriyindi;

}

else if (c->getEnd2()->getIThing()->getGroupName() == "sph" || c-
>getEnd1()->getIThing()->getGroupName() == "sph")

{

    DVect pos1(0.0);

pos2(0.0);

DVect
```

```
    quintptr ID1 = c->getEnd1()->getIThing()->getID();
quintptr ID2 = c->getEnd2()->getIThing()->getID();
    quintptr ID3 = ball->getIPiece()->getIThing()->getID();

    if (ID3==ID1)
    {
        IBall *ball2 = m->findBallWithID(ID2);

pos1 = ball->getIPieceMechanical()->getPosition();
pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

    if (q<=2.0)
    {
        w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
    }
    else
    {
        w =0.0;
    }

        const IFishParam *getvfractionsetptr = ball2->getIThing()-
>getExtra(3);
getvfraction = getvfractionsetptr->toDouble();

        sphvel = ball2->getVelocity();
        sphdens = ball2->getDensity();
    }
    else if (ID3==ID2)
    {
        IBall *ball2 = m->findBallWithID(ID1);

pos1 = ball->getIPieceMechanical()->getPosition();
```

```
pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

if (q<=2.0)
{
    w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
}
else
{
    w =0.0;
}

const IFishParam *getvfractionsetptr = ball2->getIThing()-
>getExtra(3);
getvfraction = getvfractionsetptr->toDouble();

    sphvel = ball2->getVelocity();
    sphdens = ball2->getDensity();
}
    numetvfraction += getvfraction*pow(dp,2)*w;
numetrho += sphdens*pow(dp,2)*w;
    numetvelx += sphvel.x()*pow(dp,2)*w;
    numetvely += sphvel.y()*pow(dp,2)*w;
    denomt += pow(dp,2)*w;
}
}

if (denomt ==
{
    DVect force;

    double demmass =

    force.x_ = 0.0;
    force.y_ =

    ball-

    ball-
>getIBodyMechanical()->setAppliedForce(force);
```

```

}
else
{
    IFishParam
    double ex4 =

    demvfraction =
    fluidarho =
    fluidavelx =
    fluidavely =

    double demradius
    DVect demvel =

    double vabx =
    double vaby =

    double re =
    fluidarho*demvfraction*sqrt(pow(vabx,2)+pow(vaby,2))*demradius*2.0/viscous;

    double cd;
    if (re <=
    {
        if (re ==
        {
            cd
        }
        else
        {
            cd
        }
    }
    else
    {
        cd =

    }

    double coe;
    if (demvfraction
    {
        coe =
        150.0*pow(1-
        demvfraction,2)/demvfraction*viscous/pow(demradius*2.0,2)+1.75*(1.0-
        demvfraction)*fluidarho/demradius/2.0*sqrt(pow(vabx,2)+pow(vaby,2));

```



```

}
else
{
    coe =
0.75*cd*demvfraction*(1.0-
demvfraction)/demradius/2.0*fluidarho*sqrt(pow(vabx,2)+pow(vaby,2))*pow(demvfr
action,-2.65);
}
double dragx =
double dragy =
double demrho =
double buoyy =
DVect force;
force.x_ =
force.y_ =
double demmass =
force.x_ +=
force.y_ +=
//stream<<"force
is "<<force<<endl;
ball-
>getIBodyMechanical()->setAppliedForce(force);
}
for (IIterator<IBall>
it(m->getIContainerBall());!it.atEnd();++it) {
    IBall *ball = it;
    if (ball-
double
double xvelindiy;
double
double xveltotaly = 0.0;
double
double drhodt =
0.0;
```

```

                                                                 double
acceindex;
                                                                 double acceindex;
                                                                 double
accesstotalx = 0.0;
                                                                 double
accesstotaly = 0.0;
                                                                 DVect pos1(0.0);
                                                                 DVect pos2(0.0);
                                                                 double
extforceindex;
                                                                 double
extforceyindex;
                                                                 double
extforceext = 0.0;
                                                                 double
extforceyext = 0.0;
                                                                 double denomt =
0.0;
                                                                 double
demradius;
                                                                 double ex44;
FArray<IContact *> list;
ball->getIPiece()->getContactList(&list);
                                                                 for (quintptr
i=0; i<list.size(); ++i)
                                                                 {
                                                                 IContact
*c = list.at(i);
                                                                 if
(c->getEnd2()->getIThing()->getTypeName() != "ball" || c->getEnd1()-
>getIThing()->getTypeName() != "ball") continue;
                                                                 if
(c->getEnd2()->getIThing()->getGroupName() != "particle" && c->getEnd1()-
>getIThing()->getGroupName() != "particle")
                                                                 {
                                                                 quintptr ID1 = c->getEnd1()->getIThing()->getID();
quintptr ID2 = c->getEnd2()->getIThing()->getID();
                                                                 quintptr ID3 = ball->getIPiece()->getIThing()->getID();
                                                                 if (ID3==ID1)
                                                                 {
                                                                 IBall *ball2 = m->findBallWithID(ID2);
pos1 = ball->getIPieceMechanical()->getPosition();

```

```
pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

if (q<=2.0)
{
    w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
    dwx = kc*(pos1.x()-pos2.x())/dist/h*(-4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
    dwy = kc*(pos1.y()-pos2.y())/dist/h*(-4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
}
else
{
    w = 0.0;
    dwx = 0.0;
    dwy = 0.0;
}

const IFishParam *ex222setptr = ball->getIThing()->getExtra(1);
const IFishParam *ex2222setptr = ball2->getIThing()->getExtra(1);
double ex222 = ex222setptr->toDouble();
double ex2222 = ex2222setptr->toDouble();

const IFishParam *ex333setptr = ball->getIThing()->getExtra(2);
const IFishParam *ex3333setptr = ball2->getIThing()->getExtra(2);
double ex333 = ex333setptr->toDouble();
double ex3333 = ex3333setptr->toDouble();

DVect vball1 = ball->getIPieceMechanical()->getVelocity();
DVect vball2 = ball2->getIPieceMechanical()->getVelocity();

double vabx = vball1.x()-vball2.x();
```

```
double vaby = vball1.y()-vball2.y();

double xabx = pos1.x()-pos2.x();

double xaby = pos1.y()-pos2.y();

double dens1 = ball->getIBodyMechanical()->getDensity();
double dens2 = ball2->getIBodyMechanical()->getDensity();

double visx = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vabx;

double visy = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vaby;

double anticlump =
pow(vmax/cs,2)*abs(ex333/pow(ex222,2)+ex3333/pow(ex2222,2))*pow(w/initialw,4);

if (ball2->getIThing()->getGroupName()=="sph")
{
    acceindx = -
mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+visx)*dwx;

acceindiy = -mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+visy)*dwy;

    xvelindx = eps*mass/((dens1+dens2)/2.0)*(-1.0)*vabx*w;
    xvelindiy = eps*mass/((dens1+dens2)/2.0)*(-1.0)*vaby*w;

    indidrhodt = mass*(vabx*dwx+vaby*dwy);
}

else
{
    acceindx = -
mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visx)*dwx;

acceindiy = -mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visy)*dwy;

    xvelindx = 0.0;
    xvelindiy = 0.0;

    indidrhodt = mass*(vabx*dwx+vaby*dwy);
}
}
```

```
        else if (ID3==ID2)
        {
IBall *ball2 = m->findBallWithID(ID1);

pos1 = ball->getIPieceMechanical()->getPosition();

pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

if (q<=2.0)
{
            w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
            dwx = kc*(pos1.x()-pos2.x())/dist/h*(-4.0*pow(2.0-
q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
            dwy = kc*(pos1.y()-pos2.y())/dist/h*(-4.0*pow(2.0-
q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
}
else
{
            w = 0.0;
            dwx = 0.0;
            dwy = 0.0;
}

        const IFishParam *ex222setptr = ball->getIThing()->getExtra(1);
        const IFishParam *ex2222setptr = ball2->getIThing()->getExtra(1);
double ex222 = ex222setptr->toDouble();
        double ex2222 = ex2222setptr->toDouble();

        const IFishParam *ex333setptr = ball->getIThing()->getExtra(2);
        const IFishParam *ex3333setptr = ball2->getIThing()->getExtra(2);
double ex333 = ex333setptr->toDouble();
        double ex3333 = ex3333setptr->toDouble();
}
```

```
DVect vball1 = ball->getIPieceMechanical()->getVelocity();
DVect vball2 = ball2->getIPieceMechanical()->getVelocity();

double vabx = vball1.x()-vball2.x();
double vaby = vball1.y()-vball2.y();

double xabx = pos1.x()-pos2.x();
double xaby = pos1.y()-pos2.y();

double dens1 = ball->getIBodyMechanical()->getDensity();
double dens2 = ball2->getIBodyMechanical()->getDensity();

double visx = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vabx;

double visy = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vaby;

double anticlump =
pow(vmax/cs,2)*abs(ex333/pow(ex222,2)+ex3333/pow(ex2222,2))*pow(w/initialw,4);

if (ball2->getIThing()->getGroupName()=="sph")
{
    acceindx = -
mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+visx)*dwx;

acceindiy = -mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+visy)*dwy;

    xvelindx = eps*mass/((dens1+dens2)/2.0)*(-1.0)*vabx*w;
    xvelindiy = eps*mass/((dens1+dens2)/2.0)*(-1.0)*vaby*w;

    indidrhodt = mass*(vabx*dwx+vaby*dwy);
}

else
{
    acceindx = -
mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visx)*dwx;

acceindiy = -mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visy)*dwy;

    xvelindx = 0.0;
    xvelindiy = 0.0;
```

```
        indidrhodt = mass*(vabx*dwx+vaby*dwy);
    }
}
acchetotalx += acceindex;
acchetotaly += acceindex;
drhodt += indidrhodt;
xveltotalx += xvelindex;
xveltotaly += xvelindex;
}

else if (c->getEnd2()->getIThing()->getGroupName() == "particle" || c-
>getEnd1()->getIThing()->getGroupName() == "particle")
{
    quintptr ID1 = c->getEnd1()->getIThing()->getID();
    quintptr ID2 = c->getEnd2()->getIThing()->getID();
    quintptr ID3 = ball->getIPiece()->getIThing()->getID();

    if (ID3==ID1)
    {
        IBall *ball2 = m->findBallWithID(ID2);

        demradius = ball2->getRadius();

        pos1 = ball->getIPieceMechanical()->getPosition();

        pos2 = ball2->getIPieceMechanical()->getPosition();

        dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

        double q = dist / h;

        if (q<=2.0)
        {
            w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
        }
        else
        {
```

```
        w =0.0;
    }

    extforcexindi = ball2->getAppliedForce().x();
    extforceyindi = ball2->getAppliedForce().y();

    const IFishParam *ex44setptr = ball2->getIThing()->getExtra(4);
    ex44 = ex44setptr->toDouble();
}

else if (ID3==ID2) {

IBall *ball2 = m->findBallWithID(ID1);

    demradius = ball2->getRadius();

    pos1 = ball->getIPieceMechanical()->getPosition();

    pos2 = ball2->getIPieceMechanical()->getPosition();

    dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

    double q = dist / h;

    if (q<=2.0)
    {
        w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
    }
    else
    {

        w =0.0;
    }

    extforcexindi = ball2->getAppliedForce().x();
    extforceyindi = ball2->getAppliedForce().y();

    const IFishParam *ex44setptr = ball2->getIThing()->getExtra(4);
    ex44 = ex44setptr->toDouble();
}
```



```
    extforcext -= pow(dp,2)*w*extforcexindi;
    extforceyt -= pow(dp,2)*w*extforceyindi;
    denomt += ex44;
    }
}
if (denomt ==
0.0)
{
    accetotaly
    -= 9.81;
    const
    double
    IFishParam *ex222setptr = ball->getIThing()->getExtra(1);
    double
    ex222 = ex222setptr->toDouble();
    const
    double
    IFishParam *getvfractionsetptr = ball->getIThing()->getExtra(3);
    double
    getvfraction = getvfractionsetptr->toDouble();
    rho =
    (drhodt*dt+ex222)/getvfraction;
    ball-
    >getIBodyMechanical()->setDensity(rho);
    DVect
    velnext;
    DVect
    velnow = ball->getVelocity();
    velnext.x_ = velnow.x()+accetotalx*dt;
    velnext.y_ = velnow.y()+accetotaly*dt;
    velnext.x_ += xveltotalx;
    velnext.y_ += xveltotaly;
    DVect
    posnow = ball->getIPieceMechanical()->getPosition();
    if
    (posnow.x_ >= xmax || posnow.x_ <= xmin)
    {
    velnext.x_ = 0.0;
    }
    if
    (posnow.y_ >= ymax)
    {
    velnext.y_ = 0.0;
    }
}
```

```
//stream<<"velnext is "<<velnext<<endl;

>setVelocity(velnext);

acchetotalx += extforcext/denomt/mass;
acchetotaly += extforceyt/denomt/mass;
acchetotaly -= 9.81;

IFishParam *ex222setptr = ball->getIThing()->getExtra(1);
ex222 = ex222setptr->toDouble();

IFishParam *getvfractionsetptr = ball->getIThing()->getExtra(3);
getvfraction = getvfractionsetptr->toDouble();

= (drhodt*dt+ex222)/getvfraction;

>getIBodyMechanical()->setDensity(rho);

velnext;

velnow = ball->getVelocity();

velnext.x_ = velnow.x()+acchetotalx*dt;
velnext.y_ = velnow.y()+acchetotaly*dt;

velnext.x_ += xveltotalx;
velnext.y_ += xveltotaly;

posnow = ball->getIPieceMechanical()->getPosition();

(posnow.x_ >= xmax || posnow.x_ <= xmin)

velnext.x_ = 0.0;

(posnow.y_ >= ymax)

velnext.y_ = 0.0;
```

ball-

}

else

{

const

double

const

double

double rho

ball-

DVect

DVect

DVect

if

{

}

if

{

}

```
//stream<<"velnext is "<<velnext<<endl;

                                                                 ball-
>setVelocity(velnext);
                                                                 }
                                                                 }
                                                                 for
(IIterator<IBall> it(m->getIContainerBall());!it.atEnd();++it) {
                                                                 IBall *ball = it;
                                                                 if (ball-
>getIThing()->getGroupName()!="structure") continue;
                                                                 double
                                                                 double
                                                                 double
                                                                 double
                                                                 double
                                                                 DVect pos1(0.0);
                                                                 DVect pos2(0.0);
                                                                 FArray<IContact *> list;
                                                                 ball->getIPiece()->getContactList(&list);
                                                                 for (quintptr
i=0; i<list.size(); ++i)
                                                                 {
                                                                 IContact
                                                                 if
(c->getEnd2()->getIThing()->getTypeName() != "ball" || c->getEnd1()-
>getIThing()->getTypeName() != "ball")
                                                                 {
                                                                 c->setModelName("linear");
                                                                 c->setProperty(kn, "kn");
                                                                 c->setProperty(ks, "ks");
                                                                 c->setProperty(fric, "fric");
                                                                 c->setProperty(dp_nratio, "dp_nratio");
                                                                 }
                                                                 else if (c->getEnd2()->getIThing()->getTypeName() == "particle" || c-
>getEnd1()->getIThing()->getTypeName() == "particle")
                                                                 {
                                                                 c->setModelName("linear");
                                                                 c->setProperty(kn, "kn");
                                                                 c->setProperty(ks, "ks");
                                                                 c->setProperty(fric, "fric");
                                                                 c->setProperty(dp_nratio, "dp_nratio");
```

```

}

else if (c->getEnd2()->getIThing()->getTypeName() == "sph" || c-
>getEnd1()->getIThing()->getTypeName() == "sph")
{

    quintptr ID1 = c->getEnd1()->getIThing()->getID();
quintptr ID2 = c->getEnd2()->getIThing()->getID();
    quintptr ID3 = ball->getIPiece()->getIThing()->getID();

    if (ID3==ID1)
    {
        IBall *ball2 = m->findBallWithID(ID2);

pos1 = ball->getIPieceMechanical()->getPosition();

pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

if (q<=2.0)
{
    w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
    dwx = kc*(pos1.x()-pos2.x())/dist/h*(-4.0*pow(2.0-
q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
    dwy = kc*(pos1.y()-pos2.y())/dist/h*(-4.0*pow(2.0-
q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
}
else
{
    w = 0.0;
    dwx = 0.0;
    dwy = 0.0;
}
}

```

```
const IFishParam *ex222setptr = ball->getIThing()->getExtra(1);
const IFishParam *ex2222setptr = ball2->getIThing()->getExtra(1);
double ex222 = ex222setptr->toDouble();
double ex2222 = ex2222setptr->toDouble();

const IFishParam *ex333setptr = ball->getIThing()->getExtra(2);
const IFishParam *ex3333setptr = ball2->getIThing()->getExtra(2);
double ex333 = ex333setptr->toDouble();
double ex3333 = ex3333setptr->toDouble();

DVect vball1 = ball->getIPieceMechanical()->getVelocity();
DVect vball2 = ball2->getIPieceMechanical()->getVelocity();

double vabx = vball1.x()-vball2.x();
double vaby = vball1.y()-vball2.y();

double xabx = pos1.x()-pos2.x();
double xaby = pos1.y()-pos2.y();

double dens1 = ball->getIBodyMechanical()->getDensity();
double dens2 = ball2->getIBodyMechanical()->getDensity();

double visx = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vabx;

double visy = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vaby;

double anticlump =
pow(vmax/cs,2)*abs(ex333/pow(ex222,2)+ex3333/pow(ex2222,2))*pow(w/initialw,4);

acceindx = -
mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visx)*dwx;

acceindiy = -mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visy)*dwy;
}

else if (ID3==ID2)
{
IBall *ball2 = m->findBallWithID(ID1);
```

```
pos1 = ball->getIPieceMechanical()->getPosition();

pos2 = ball2->getIPieceMechanical()->getPosition();

dist = sqrt(pow((pos1.x()-pos2.x()),2)+pow((pos1.y()-pos2.y()),2));

double q = dist / h;

if (q<=2.0)
{
    w = kc*(pow(2.0-q,4)*(1.0+2.0*q));
    dwx = kc*(pos1.x()-pos2.x())/dist/h*(-4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
    dwy = kc*(pos1.y()-pos2.y())/dist/h*(-4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0);
}
else
{
    w = 0.0;
    dwx = 0.0;
    dwy = 0.0;
}

const IFishParam *ex222setptr = ball->getIThing()->getExtra(1);
const IFishParam *ex2222setptr = ball2->getIThing()->getExtra(1);
double ex222 = ex222setptr->toDouble();
double ex2222 = ex2222setptr->toDouble();

const IFishParam *ex333setptr = ball->getIThing()->getExtra(2);
const IFishParam *ex3333setptr = ball2->getIThing()->getExtra(2);
double ex333 = ex333setptr->toDouble();
double ex3333 = ex3333setptr->toDouble();

DVect vball1 = ball->getIPieceMechanical()->getVelocity();
DVect vball2 = ball2->getIPieceMechanical()->getVelocity();
```

```
double vabx = vball1.x()-vball2.x();
double vaby = vball1.y()-vball2.y();

double xabx = pos1.x()-pos2.x();
double xaby = pos1.y()-pos2.y();

double dens1 = ball->getIBodyMechanical()->getDensity();
double dens2 = ball2->getIBodyMechanical()->getDensity();

double visx = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vabx;

double visy = mass*(viscous+viscous)/dens1/dens2/dist/h*(-
4.0*pow(2.0-q,3)*(1.0+2.0*q)+pow(2.0-q,4)*2.0)*vaby;

double anticlump =
pow(vmax/cs,2)*abs(ex333/pow(ex222,2)+ex3333/pow(ex2222,2))*pow(w/initialw,4);

acceindx = -
mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visx)*dwx;

acceindiy = -mass*(ex333/pow(ex222,2)+ex3333/pow(ex2222,2)+0.2*visy)*dwy;
}
acctotalx += acceindx;
acctotaly += acceindiy;
}

acctotaly -= 9.81;
double
fmass = ball->getIBodyMechanical()->getMass();
DVect
force;
force.x_ = fmass*(acctotalx);
force.y_ = fmass*(acctotaly);
ball-
>getIBodyMechanical()->setAppliedForce(force);
}
}
}
break;
default:
throw std::runtime_error("Illegal ID in
FishSPH::get().");
}
}
```