Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Applied Mathematical Modelling 29 (2005) 1252–1270

APPLIED MATHEMATICAL MODELLING

www.elsevier.com/locate/apm

Modeling incompressible flows using a finite particle method

M.B. Liu [a,*,1], W.P. Xie [b], G.R. Liu [a]

[a] *Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore*

[b] *Department of Mathematics, Shaoyang University, Shaoyang, Hunan 422000, PR China*

Received 1 October 2003; received in revised form 1 May 2005; accepted 25 May 2005
Available online 12 July 2005

Abstract

This paper describes the applications of a finite particle method (FPM) to modeling incompressible flow problems. FPM is a meshfree particle method in which the approximation of a field variable and its derivatives can be simultaneously obtained through solving a pointwise matrix equation. A set of basis functions is employed to obtain the coefficient matrix through a sequence of transformations. The finite particle method can be used to discretize the Navier–Stokes equation that governs fluid flows. The incompressible flows are modeled as slightly compressible via specially selected equations of state. Four numerical examples including the classic Poiseuille flow, Couette flow, shear driven cavity and a dam collapsing problem are presented with comparisons to other sources. The numerical examples demonstrate that FPM is a very attractive alternative for simulating incompressible flows, especially those with free surfaces, moving interfaces or deformable boundaries.

Keywords: Meshless/meshfree methods; Finite particle method; Smoothed particle hydrodynamics; Basis function; Incompressible flow

[*] Corresponding author. Tel./fax: +1 208 5265969.
E-mail address: moubl@inel.gov (M.B. Liu).

[1] Present address: Idaho National Engineering and Environmental Laboratory, P.O. Box 1625, MS 2025, Idaho Falls, ID 83415-2025, USA.

0307-904X/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2005.05.003

1. Introduction

Simulation of incompressible flows is a very important topic in computational fluid dynamics (CFD). Traditionally grid-based numerical methods such as finite element method (FEM), finite difference method (FDM) and finite volume method (FVM) are used to simulate incompressible flows. FEM employs a properly selected interpolation function on usually an unstructured mesh to generate the system matrix equations which are derived using the variation principle or some kind of Weighted Residual Method (WRM) such as Galerkin Method [1]. FDM and FVM usually involve solving the pressure Poisson equation (either original or corrected) in a staggered grid [2] or collocated grid [3]. FDM and FVM have been dominating the field of CFD for quite a along period of time.

Dealing with moving interfaces and free surfaces (or moving boundaries in all) in an Eulerian grid fixed on the problem space is generally a tough task for FDM and FVM. The PIC (particle-in-cell) [4] method uses particles to locate the moving boundaries and finite difference to solve the governing equations. The PIC method and another similar approach, MAC (marker-and-cell) [5], have been widely used to solve complex computational fluid dynamic problems including reactive flows, multi-material flows, multi-phase flows and flows with spatial discontinuities [6]. Though PIC and MAC are flexible and robust, they are quite complicated in programming and need expensive computational effort especially for three-dimensional cases [7]. More recent progresses in treating moving boundaries usually involve some kind of boundary capturing techniques such as VOF [8] and level-set [9]. The VOF method solves an additional partial differential equation for the filled fraction of each control volume besides the conservation equations. The level-set method also needs to solve an additional level-set function except for the conservation equations.

Recently growing interests have been focused on the so-called meshless methods or meshfree methods as alternatives for traditional grid-based methods. The initial motivation of the meshless methods was to modify the internal structure of the grid-based FDM and FEM so as to become more adaptive, versatile and robust. Since the emergence of the smoothed particle hydrodynamics (SPH) [10,11] for solving astrophysical problems in open space, a large number of new meshless methods have been proposed for analyzing solids and structures as well as fluid flows. Typical of them are finite point method [12,13], diffuse element method (DEM) [14], element free Galerkin (EFG) method [15], reproduced kernel particle method (RKPM) [16], HP-cloud method [17], free mesh method [18], meshless local Petrov–Galerkin (MLPG) method [19], point interpolation method (PIM) [20] and many others. Belytschko et al. [21], Li and Liu [22], and Liu and Liu [23] provided detailed reviews on the meshless methods. These meshless methods share some common features, but are different in the approximation and implementation process. The key idea (and also the objective) of these meshless methods is to seek accurate and stable numerical solutions for integral equations or partial differential equations (PDEs) through using a set of nodes (or particles) rather than any grid. Meshless or meshfree methods have been found to have special advantages on problems to which the conventional grid-based methods are difficult to be applied. These generally include problems with free surfaces, deformable boundaries, moving interfaces (difficult for FDM), large deformations (difficult for FEM), complex mesh generation, mesh adaptivity, and multi-scale resolution (difficult for both FDM and FEM) [23].

This paper presented the application of a finite particle method (FPM) to modeling incompressible flows. Similar to SPH, FPM is a meshless method in which a set of Lagrangian particles are used to represent the state of the simulated system and these particles can move according to forces calculated from particle–particle interactions. FPM restores the particle inconsistency that is inherently associated with the conventional SPH, and can have better accuracy than the conventional SPH. FPM can be regarded as a modified or an enhanced version of SPH.

The paper is organized as follows. The approximation scheme in FPM was described in Section 2 together with some discussions on the similarities and differences of several modified versions of SPH. Section 3 addressed the FPM discretization of the Navier–Stokes equation, and Section 4 described the treatment of modeling incompressible flows as slightly compressible flows. In Section 5, four numerical examples including the classic Poiseuille flow, Couette flow, shear driven cavity and a dam collapsing problem were presented to demonstrate the good performance of the finite particle method with comparisons to other sources. The paper ended in Section 6 with some conclusions.

2. Finite particle method (FPM)

2.1. Approximations in FPM

Considering a problem domain Ω filled with a set of particles (usually arbitrarily distributed, see Fig. 1 for a two-dimensional illustration). These particles can either be the central particles initially generated using existing mesh generation tools or concentrated particles initially generated using some kind of space discretization model such as the particle-fill model in AUTODYN [24]. The state of the system is represented by these particles, each associated with field properties. These particles can be used not only for integration, interpolation or differencing, but also for representing the material as mass particles. One may regard these particles as the mass centers of the corresponding subsections of the material. The volume of a subsection is lumped on the

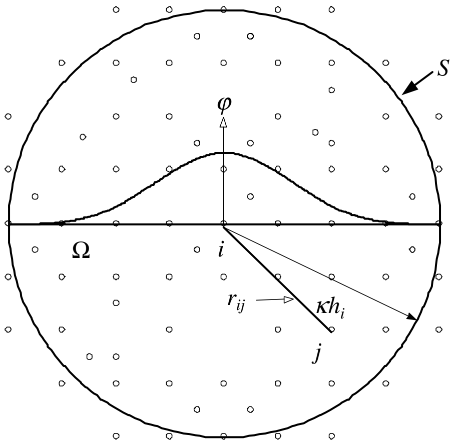

Fig. 1. FPM approximation in a two-dimensional problem domain Ω with a surface S. φ is a basis function that is used to approximate the field variables at particle i through using the particles j within the support domain represented by κh_i.

corresponding particle. Therefore one particle i is associated with a fixed lumped volume ΔV_i without fixed shape. If the particle mass and density are concerned, the lumped volume can also be replaced by the corresponding mass to density ratio m_i/ρ_i. These particles can be fixed in an Eulerian frame or move in a Lagrangian frame. Since an approximation is to be carried out by using only this set of finite number of particles, the method to be discussed is termed as finite particle method (FPM).

Numerical simulations usually involve approximation of the values of a function and its derivatives at a certain point. Performing Taylor series expansion at a nearby point $\boldsymbol{x}_i = \{x_i, y_i, z_i\}$ and retaining the first order derivatives, a sufficiently smooth function $f(\boldsymbol{x})$ at point $\boldsymbol{x} = \{x, y, z\}$ can be expressed as follows:

$$f(\boldsymbol{x}) = f_i + (\boldsymbol{x}^\alpha - \boldsymbol{x}_i^\alpha) f_{i,\alpha} + r((\boldsymbol{x} - \boldsymbol{x}_i)^2), \tag{1}$$

where α is the dimension index repeated from 1 to 3 (or from x to z). $r((\boldsymbol{x} - \boldsymbol{x}_i)^2)$ is the remainder of the expansion. f_i, and $f_{i,\alpha}$ are defined as

$$f_i = f(\boldsymbol{x}_i), \tag{2}$$

$$f_{i,\alpha} = f_\alpha(\boldsymbol{x}_i) = (\partial f / \partial \boldsymbol{x}^\alpha)_i. \tag{3}$$

Multiplying both sides of Eq. (1) with a weight function $\varphi_1(\boldsymbol{x} - \boldsymbol{x}_i)$ and integrating over the problem space Ω can yield the following equation:

$$\int_\Omega f(\boldsymbol{x}) \varphi_1(\boldsymbol{x} - \boldsymbol{x}_i) \mathrm{d}\boldsymbol{x} = f_i \int_\Omega \varphi_1(\boldsymbol{x} - \boldsymbol{x}_i) \mathrm{d}\boldsymbol{x} + f_{i,\alpha} \int_\Omega (\boldsymbol{x}^\alpha - \boldsymbol{x}_i^\alpha) \varphi_1(\boldsymbol{x} - \boldsymbol{x}_i) \mathrm{d}\boldsymbol{x} + r((\boldsymbol{x} - \boldsymbol{x}_i)^2). \tag{4}$$

If assuming φ_1 to be defined over the entire domain, the integration in Eq. (4) needs to be carried out over the entire problem space, and can be quite time-consuming. One usual assumption is that a field variable at point $\boldsymbol{x}_i$ is only strongly influenced by the field variables at nearby points and that the influence of the field variables at points far away from point $\boldsymbol{x}_i$ is very weak and hence can be neglected. This can be implemented by defining $\varphi_1(\boldsymbol{x} - \boldsymbol{x}_i)$ as a function $\varphi_1(\boldsymbol{x} - \boldsymbol{x}_i, h)$ that has a compact support domain, which is usually characterized by a circle (in 2D) or a sphere (in 3D) with a radius of κh, in which κ is a constant scalar factor, and h is a length characterizing the support domain. As such the global integration at a certain point $\boldsymbol{x}_i$ can be converted into a local integration, and hence the computational effort can be reduced.

Since the points distributed in the problem space are actually particles, each occupying individual lumped volume, Eq. (4) can be numerically approximated by summation over the particles surrounding point $\boldsymbol{x}_i$ as follows:

$$\sum_{j=1}^{N} f(\boldsymbol{x}_j) \varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h) \Delta V_j = f_i \sum_{j=1}^{N} \varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h) \Delta V_j + f_{i,\alpha} \sum_{j=1}^{N} (\boldsymbol{x}_j^\alpha - \boldsymbol{x}_i^\alpha) \varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h) \Delta V_j, \tag{5}$$

where N is the number of particles within the support domain of particle i. The remainder term $r((\boldsymbol{x} - \boldsymbol{x}_i)^2)$ in Eq. (4) is omitted in Eq. (5) for the sake of conciseness. The summation over the particles is illustrated in Fig. 1.

Eq. (5) can be rewritten as the following equation at point $\boldsymbol{x}_i$:

$$B_{1i} = A_{1ki}F_{ki}, \tag{6}$$

where

$$F_{ki} = [f_i \quad f_{i,\alpha}]^{\mathrm{T}}, \tag{7}$$

$$B_{1i} = \sum_{j=1}^{N} f(\boldsymbol{x}_j)\varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j, \tag{8}$$

and

$$A_{1ki} = \left[\sum_{j=1}^{N} \varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j \quad \sum_{j=1}^{N} (\boldsymbol{x}_j^{\alpha} - \boldsymbol{x}_i^{\alpha})\varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j \right]. \tag{9}$$

Corresponding to one-, two-, and three-dimensional cases, there are 1 function value, and 1, 2 and 3 first order derivatives that will be approximated. It is clear that k in Eqs. (7), (9) and (6) is 2, 3, and 4 respectively corresponding to one-, two-, and three-dimensional cases. To calculate the function value, and the first order derivatives at $\boldsymbol{x}_i$, 1, 2, and 3 other equations similar to Eq. (6) are required. Therefore in one-, two-, and three-dimensional cases, totally 2, 3, and 4 functions ($\varphi_M(\boldsymbol{x} - \boldsymbol{x}_i, h)$, $M = 2$, 3, or 4) are required in order to approximate the function value and the first order derivative. These functions form a set of basis functions used for approximating the function value, and its first derivatives.

In summary, multiplying a set of basis functions on both sides of Eq. (1), integrating over the problem domain, summing over the nearest particles within the local support domain of particle i, a set of local matrix equation can be produced to approximate the function value as well as the first derivatives at particle i. The matrix equations at particle i at can be written as

$$B_{Mi} = A_{Mki}F_{ki}, \tag{10}$$

where B_{Mi} and A_{Mki} are similar to what defined in Eqs. (8) and (9) with a constant subscript M.

Eq. (10) is the basis of the finite particle method and will be used to approximate a function value and its derivatives for a field variable. FPM actually generates a local matrix at each particle and solving the function value and its derivatives involves matrix inversion. If the coefficient matrix A is not singular, these M equations can determine a unique set of solutions at particle i for the M unknowns in vector F. The approximations expressed in Eq. (10) have second accuracy since the function and the first order derivatives are retained in the approximation process.

The governing equations in CFD or computational solid dynamics (CSD) may involve higher order derivatives. There are basically two different approaches to approximate higher order derivatives. The first one is to directly calculate the high order derivatives. This can be implemented in a similar procedure by retaining the higher order derivatives when expanding $f(\boldsymbol{x})$ at $\boldsymbol{x}_i$ in Eq. (1), and obtaining the increased number of unknowns using increased number of functions like $\varphi_M(\boldsymbol{x} - \boldsymbol{x}_i, h)$. The resultant local matrix equation contains more terms related to high order derivatives. Therefore this approach is more computationally expensive, and is not preferable especially for large-scale simulations. A feasible alternative is to calculate the high order derivatives indirectly as the first order derivative of the lower order derivatives. This is a nested approximation on the first order derivatives with the same order accuracy, and is to be used in this paper when approximating strain rate.

2.2. More discussions[2]

The smoothed particle hydrodynamics (SPH) method is the oldest meshfree particle methods [10,11]. It has special advantages over other meshless methods since the nodes in SPH not only act as field points for integration and approximation, but also act as mass-lumped (m_i/ρ_i), Lagrangian particles that can freely move in the problem space according to forces calculated from particle–particle interactions. Since its invention to solve three-dimensional astrophysical problems, different versions of SPH have been comprehensively investigated with wide applications to a series of challenging problems [25–31].

The smoothing function W in SPH methods is usually assumed to be symmetric ($\int_\Omega (x^\alpha - x_i^\alpha) W \, \mathrm{d}\boldsymbol{x} = 0$), and satisfy the normalization condition ($\int W(\boldsymbol{x} - \boldsymbol{x}_i, h)\mathrm{d}\boldsymbol{x} = 1$) [25] for interior regions and problems with infinite boundaries (e.g. problems in astrophysics). In Eq. (4), if ignoring the terms related to derivatives, replacing the weight function $\varphi_1(\boldsymbol{x} - \boldsymbol{x}_i, h)$ with the smoothing function $W(\boldsymbol{x} - \boldsymbol{x}_i, h)$, and re-arranging the terms, we can obtain the following equation:

$$f_i = \frac{\int_\Omega f(\boldsymbol{x}) W(\boldsymbol{x} - \boldsymbol{x}_i, h)\mathrm{d}\boldsymbol{x}}{\int_\Omega W(\boldsymbol{x} - \boldsymbol{x}_i, h)\mathrm{d}\boldsymbol{x}}. \tag{11}$$

Comparing Eq. (11) with the conventional SPH kernel approximation ($f_i = \int_\Omega f(\boldsymbol{x}) W(\boldsymbol{x} - \boldsymbol{x}_i, h)\,\mathrm{d}\boldsymbol{x}$, [25]), it is found that for interior regions, the *kernel approximations* (continuous form approximation) in the conventional SPH and in Eq. (11) are actually the same due to the satisfaction of the normalization condition. For boundary regions, since the integral of the smoothing function is truncated by the boundary, the normalization condition cannot be satisfied ($\int W(\boldsymbol{x} - \boldsymbol{x}_i, h)\,\mathrm{d}\boldsymbol{x} \neq 1$). Assuming the integration of the smoothing function in the boundary regions to be unity as in interior regions is the essential reason that the conventional SPH kernel approximation has lower order accuracy in boundary regions than the interior regions. Note that the conventional SPH kernel approximation was usually claimed to have second order accuracy considering the symmetric and normalization properties of the smoothing function. This is true only for interior regions. For boundary regions, the conventional SPH kernel approximation does not have first order accuracy.

The *particle approximation* (discretized form) for Eq. (11) is

$$f_i = \frac{\sum_{j=1}^N f(\boldsymbol{x}_j) W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j}{\sum_{j=1}^N W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j}. \tag{12}$$

The approximation expressed in Eq. (12) is of first order accuracy since no derivative terms are retained. If assuming $\sum_{j=1}^N W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j = 1$, the particle approximation in the conventional SPH can easily obtained as

$$f_i = \sum_{j=1}^N f_j W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j. \tag{13}$$

However, the discretized form of normalization condition $\sum_{j=1}^N W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j = 1$ is valid only for very few simple cases, but not valid for general applications with randomly distributed

[2] Through personal communications, the authors found that Prof. Batra and his colleague, Zhang, had similar ideas to apply Taylor series expansion to SPH function approximation. They submitted their work for publication nearly at the same time as this article [43,44].

particles and finite boundaries [32]. Hence the conventional SPH particle approximation for a function does not have first order accuracy even for particles in interior regions.

The particle approximation of the first order derivatives in the conventional SPH can be obtained by simply replacing the smoothing function W with the corresponding derivatives of W as follows:

$$f_{i,\alpha} = \sum_{j=1}^{N} f_j W_{j,\alpha}(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j. \tag{14}$$

Similarly, the conventional SPH particle approximations of the first order derivative also do not have first order accurate. This is the fundamental reason that the conventional SPH method has comparatively low accuracy.

For some special problems where the particles do not move and act only as interpolation points, the accuracy of SPH approximations can be enhanced to some extent through selecting a suitable smoothing function, optimizing the smoothing length h, and evenly distributing the SPH particles, as discussed in [32]. However, for general applications where large transient deformation occurs and the particles also act as moving Lagrangian particles, these accuracy-enhancing measures are not efficient. Therefore since the invention of the conventional SPH, various modifications have been developed to improve its accuracy. The symmetrization and anti-symmetrization in some SPH formulations [25] are early efforts in trying to improve particle approximation accuracy, but often fail to do so especially for transient problems with large deformations.

Liu et al. [16] developed a reproducing kernel particle method by correcting the smoothing function W so as to reproduce consistency conditions. Bonet and Kulasegaram [33] proposed a corrected smoothed particle hydrodynamics (CSPH), in which the smoothing function W can be corrected or reconstructed to enforce the consistency condition, and the node integration was corrected to enable further improvement of the accuracy. Dilts [34] presented a moving least square particle hydrodynamics (MLSPH), which replaced the conventional SPH interpolant with the moving least square interpolant. RKPM, CSPH and MLSPH all involve the correction or reconstruction of the conventional symmetric, non-negative smoothing function, and therefore reproduce a function and its derivatives to desired accuracy. The reconstructed smoothing function was negative in some parts of the region, which can leads to unphysical representation of some field variables, such as negative density, negative energy that can lead to breakdown of the entire computation when simulating hydrodynamic problems [23].

Chen and Beraun [35] developed a generalized or corrected smoothed particle hydrodynamics method (CSPM) for nonlinear dynamic problems. The particle approximation for a function in CSPM is expressed in Eq. (12) and has first order accuracy. Once f_i is obtained, the approximation of first order derivatives in CSPM can be obtained by retaining terms up to first order derivatives in Eq. (1), multiplying both sides of Eq. (1) with the first order derivatives, $W_{i,\alpha}$, of the smoothing function W, and solving the following resultant coupled equations:

$$f_{i,\alpha} \sum_{j=1}^{N} (\boldsymbol{x}_j^{\alpha} - \boldsymbol{x}_i^{\alpha}) W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j = \sum_{j=1}^{N} f(\boldsymbol{x}_j)\varphi_1(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j - f_i \sum_{j=1}^{N} W(\boldsymbol{x}_j - \boldsymbol{x}_i, h)\Delta V_j. \tag{15}$$

Similarly, the approximations of the first order derivatives in CSPM have first order accuracy.

It is seen that FPM, SPH and CSPM all are meshless method in which particles with lumped volumes are used to represent the state of a system, and form a frame for interpolation, differencing or integration in a certain approximation. They all can be used as Lagrangian methods in which the particles can move freely according to forces calculated from particle–particle interactions. FPM and CSPM, as modified versions of SPH, do not involve reconstruction or correction of the conventional SPH smoothing function.

However, the differences of FPM from SPH and CSPM are obvious.

1. FPM uses a set of basis function to approximate the function value and its derivatives, whereas SPH and CSPM employ the smoothing function and its derivatives to approximate a function value and the corresponding derivatives. The smoothing functions in SPH and CSPM should have some special properties as described in [36]. However, the basis functions in FPM are more general. Any set of functions, which do not lead to a singular coefficient matrix A, can be used as basis functions. Therefore the smoothing function and its specific derivatives actually can be one choice as a suitable set of basis functions. More discussions on the selection of basis function can be found in [37].
2. SPH particle approximations (Eqs. (13) and (14)) do not have first order accuracy, and CSPM particle approximations (Eqs. (12) and (15)) have first order accuracy. While the particle approximations in FPM have second order accuracy if terms up to first order derivatives are retained.
3. In SPH, the approximation of a function and its derivatives are solved in parallel. In CSPM, the approximation of a function (Eq. (12)) is first calculated, and then the approximations of the first order derivatives are solved simultaneously through solving the coupled expressions in Eq. (15). In FPM, a function and all its derivatives are calculated at the same time through solving the coupled local matrix equation.
4. Since FPM and CSPM involve solving the coupled, local matrix equations for each particle, they are computationally more expensive than SPH.

3. Equations of motion

The basic governing equations of fluid dynamics are based on the three fundamental physical laws of conservation, (1) conservation of mass, (2) conservation of momentum, and (3) conservation of energy. In a Lagrangian frame, the governing equations for incompressible flows can be written as follows:

$$\begin{cases} \dfrac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho \dfrac{\partial \boldsymbol{v}^{\beta}}{\partial \boldsymbol{x}^{\beta}}, \\ \dfrac{\mathrm{D}\boldsymbol{v}^{\alpha}}{\mathrm{D}t} = \dfrac{1}{\rho} \dfrac{\partial \sigma^{\alpha\beta}}{\partial \boldsymbol{x}^{\beta}}, \end{cases} \tag{16}$$

where the scalar density ρ, and internal energy e, the velocity component $\boldsymbol{v}^{\alpha}$, and the total stress tensor $\sigma^{\alpha\beta}$ are the dependent variables. The spatial coordinates $\boldsymbol{x}^{\alpha}$ and time t are the independent variables. The summation in Eq. (16) is taken over repeated indices, while the total time derivatives are taken in the moving Lagrangian frame. For incompressible flow problems, the energy equation can be removed from Eq. (16).

The total stress tensor $\sigma^{\alpha\beta}$ in Eq. (16) is made up of two parts, one part of isotropic pressure p and the other part of viscous stress τ.

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \tau^{\alpha\beta}, \tag{17}$$

where $\delta_{\alpha\beta} = 1$ if $\alpha = \beta$, and $\delta_{\alpha\beta} = 0$ if $\alpha \neq \beta$.

The pressure can be calculated through an equation of state, $p = p(\rho, e)$. For Newtonian fluids, the viscous stress should be proportional to the strain rate denoted by ε through the dynamic viscosity μ.

$$\tau^{\alpha\beta} = \mu\varepsilon^{\alpha\beta}, \tag{18}$$

where

$$\varepsilon^{\alpha\beta} = \frac{\partial \boldsymbol{v}^{\beta}}{\partial \boldsymbol{x}^{\alpha}} + \frac{\partial \boldsymbol{v}^{\alpha}}{\partial \boldsymbol{x}^{\beta}} - \frac{2}{3}(\nabla \cdot \boldsymbol{v})\delta^{\alpha\beta}. \tag{19}$$

Discretizing Eq. (16) using FPM, a set of corresponding equations of motion can be produced. If the angle brackets ⟨⟩ are used to represent the approximations of the derivatives present in Eq. (16), after linearization, the discretized equations of motion at particle i can be obtained as

$$\begin{cases} \left\langle \dfrac{\mathrm{D}\rho}{\mathrm{D}t} \right\rangle_i = -\rho_i \left\langle \dfrac{\partial \boldsymbol{v}^{\beta}}{\partial \boldsymbol{x}^{\beta}} \right\rangle_i, \\ \left\langle \dfrac{\mathrm{D}\boldsymbol{v}^{\alpha}}{\mathrm{D}t} \right\rangle_i = -\dfrac{1}{\rho_i} \left\langle \dfrac{\partial p}{\partial \boldsymbol{x}^{\alpha}} \right\rangle_i + \dfrac{\mu}{\rho_i} \left\langle \dfrac{\partial \varepsilon^{\alpha\beta}}{\partial \boldsymbol{x}^{\beta}} \right\rangle_i. \end{cases} \tag{20}$$

It is seen that the density change can be obtained by simply replacing f in Eq. (10) with $\boldsymbol{v}^{\beta}$ and calculate the derivative of the velocity. Calculation of velocity change is slightly complicated since the viscous part needs nested approximation. $\left\langle \frac{\partial p}{\partial \boldsymbol{x}^{\alpha}} \right\rangle_i$ can be easily obtained by replacing f in Eq. (10) with p and calculate the derivative of p. $\left\langle \frac{\partial \varepsilon^{\alpha\beta}}{\partial \boldsymbol{x}^{\beta}} \right\rangle_i$ involves nested approximations in which $\varepsilon^{\alpha\beta}$ can be approximated first as

$$\langle \varepsilon^{\alpha\beta} \rangle_i = \left\langle \frac{\partial \boldsymbol{v}^{\beta}}{\partial \boldsymbol{x}^{\alpha}} \right\rangle_i + \left\langle \frac{\partial \boldsymbol{v}^{\alpha}}{\partial \boldsymbol{x}^{\beta}} \right\rangle_i - \frac{2}{3}\langle (\nabla \cdot \boldsymbol{v}) \rangle_i \delta^{\alpha\beta}. \tag{21}$$

The leapfrog method is used for its low requirement on memory storage and its computational efficiency. In the leapfrog method, the particle velocities and positions are offset by a half time step when integrating the equations of motion. At the end of the first time step (t_0), the change in density and velocity are used to advance the density and velocity at half a time step, while the particle positions are advanced in a full time step.

$$\begin{cases} t = t_0 + \Delta t, \\ \rho_i(t_0 + \Delta t/2) = \rho_i(t_0) + \frac{\Delta t}{2}\,\mathrm{D}\rho_i(t_0), \\ \boldsymbol{v}_i(t_0 + \Delta t/2) = \boldsymbol{v}_i(t_0) + \frac{\Delta t}{2}\,\mathrm{D}\boldsymbol{v}_i(t_0), \\ \boldsymbol{x}_i(t_0 + \Delta t) = \boldsymbol{x}_i(t_0) + \Delta t \cdot \boldsymbol{v}_i(t_0 + \Delta t/2). \end{cases} \tag{22}$$

In order to keep the calculations consistent at each subsequent time step, at the start of each subsequent time step, the density and velocity of each particle need to be predicted at half a time step to coincide the position.

$$\begin{cases} \rho_i(t) = \rho_i(t - \Delta t/2) + \frac{\Delta t}{2}\,\mathrm{D}\rho_i(t - \Delta t), \\ \boldsymbol{v}_i(t) = \boldsymbol{v}_i(t - \Delta t/2) + \frac{\Delta t}{2}\,\mathrm{D}\boldsymbol{v}_i(t - \Delta t). \end{cases} \tag{23}$$

At the end of the subsequent time step, the particle density, velocity and position are advanced in the standard leapfrog scheme

$$\begin{cases} t = t + \Delta t, \\ \rho_i(t + \Delta t/2) = \rho_i(t - \Delta t/2) + \Delta t \cdot D\rho_i(t), \\ \boldsymbol{v}_i(t + \Delta t/2) = \boldsymbol{v}_i(t - \Delta t/2) + \Delta t \cdot D\boldsymbol{v}_i(t), \\ \boldsymbol{x}_i(t + \Delta t) = \boldsymbol{x}_i(t) + \Delta t \cdot \boldsymbol{v}_i(t + \Delta t/2). \end{cases} \tag{24}$$

4. Equation of state

Similar to the approach by Monaghan [38] and Morris et al. [39] to model incompressible flow in SPH, in FPM, the incompressible flows are also treated as slightly compressible flows using a specially designed equation of state rather than the actual equation of state [40–42].

One possible choice of the equation of state is

$$p = B\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right), \tag{25}$$

where γ is a constant taken as $\gamma = 7$ in most circumstances. ρ_0 is the reference density. B is a problem dependent parameter, which determines the sound speed and therefore sets a limit for the maximum change of the density, as discussed by Monaghan [38]. In most circumstances, B can be taken as the initial pressure. The subtraction of 1 in equation can remove the boundary effects for free surface flows [39]. It can be seen that a small oscillation in density may result in a large variation in pressure. Batchelor [40] and Monaghan [38] applied this equation of state for water to model free surface flows.

Another equation of state is

$$p = c^2\rho, \tag{26}$$

where c is the sound speed. Morris et al. [39] used this equation of state in modeling low Reynolds number incompressible flows using SPH.

The sound speed is an important factor that deserves careful consideration. If the actual sound speed (e.g. 1480 m/s for water under standard pressure and temperature) is employed, the real fluid is approximated as an ideally incompressible fluid. Monaghan [38] gave a relationship between density variation δ and Mach number or sound speed

$$\delta = \frac{\Delta\rho}{\rho_0} = \frac{|\rho - \rho_0|}{\rho_0} = \frac{V_{\mathrm{b}}^2}{c^2} = M^2, \tag{27}$$

where V_{b} and M are the fluid bulk velocity and Mach number respectively. Eq. (27) determines the selection of sound speed once the density variation range is given. Since the actually sound speed is very large, and the corresponding Mach number is very small, the density variation δ will be nearly negligible. Therefore, in order to approximate the real fluid as a quasi-incompressible or slightly compressible fluid, a much smaller value than the actual sound speed should be used. This sound speed value, on one hand, should be large enough so that the behavior of the quasi-compressible fluid is sufficiently close to the real fluid; on the other hand, should be small enough so that the time step is increased to an acceptable value. Considering the balance of the time step and the incompressible behavior of the quasi-incompressible fluid, there should be an optimal value for the sound speed. From Eq. (27), it can be seen that in modeling a real fluid as a quasi-compressible fluid, the sound speed to be chosen is closely related to the bulk velocity of the flow. Apart from the bulk velocity, the pressure field also needs to be well estimated when choosing the sound speed.

Morris et al. [39], through considering the balance of pressure, viscous force and body force, proposed an estimate for the sound speed. He argued that the square of the sound speed should be comparable with the largest value of V_{b}^2/δ, $\upsilon V_{\mathrm{b}}/\delta L$, and FL/δ, i.e.

$$c^2 = \max\left(\frac{V_{\mathrm{b}}^2}{\delta}, \frac{\upsilon V_{\mathrm{b}}}{\delta L}, \frac{FL}{\delta}\right), \tag{28}$$

where $\upsilon(\upsilon = \mu/\rho)$ is the kinetic viscosity, F is the magnitude of the body force, L is the characteristic length scale.

In this paper, both these two equations of state will be used. Eq. (26) will be used in the simulation of the Poiseuille flow, Couette flow and shear driven cavity flow. Eq. (25) will be used to model the dam collapsing problem.

5. Numerical examples

In this section, FPM was applied to simulate incompressible flows with four numerical examples including Poiseuille flow, Couette flow, shear driven cavity, and a dam collapsing problem. The first three examples were low Reynolds number incompressible flows and the last example was a usual benchmark of incompressible flows with free surfaces.

Since the examples involve comparisons with SPH that uses a smoothing function, the same smoothing function and its derivatives were also used in FPM as the basis functions to approximate the function and the corresponding derivatives. The widely used cubic spline function [25] was chosen here as the smoothing function. If defining $R_{ij} = r_{ij}/h = |\boldsymbol{x}_j - \boldsymbol{x}_i|/h$, the cubic spline function can be written as

$$W_{ij} = \alpha_d \times \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3 & 0 \leqslant R < 1, \\ \frac{1}{6}(2-R)^3 & 1 \leqslant R < 2, \\ 0 & R \geqslant 2, \end{cases} \tag{29}$$

where $\alpha_d = 1/h$ and $15/7\pi h^2$ in one- and two-dimensional spaces respectively.

5.1. Poiseuille flow

The Poiseuille flow involves flow between two parallel, stationary and infinite plates placed at $y = 0$ and $y = l$. The initially stationary fluid is driven by a body force F (e.g., a pressure difference or an external force), gradually flows between the two plates, and finally arrives at a steady state. In this example, $l = 10^{-3}$ m, $\rho = 10^3$ kg/m^3, the kinetic viscosity $\upsilon = 10^{-6}$ m^2 /s. The body force F determines the peak fluid velocity, which in turn determines the Reynolds number. Here two cases were considered, one with $F = 2 \times 10^{-4}$ m/s^2, the other with $F = 8 \times 10^{-4}$ m/s^2. Therefore the resultant peak fluid velocities were $\boldsymbol{v}_0 = 2.5 \times 10^{-5}$ m/s and $\boldsymbol{v}_0 = 1.0 \times 10^{-4}$ m/s, which correspond to Reynolds numbers of $Re = 0.025$ and $Re = 0.1$ respectively. The sound speed was taken as $c = 2.5 \times 10^{-4}$ m/s and $c = 1.0 \times 10^{-3}$ m/s, therefore the Mach numbers for the two case were $M = 0.1$.

The solid walls were modeled using virtual or ghost particles [23] in SPH and in FPM. The virtual particles in SPH act to implement the boundary conditions, and can help to improve the boundary accuracy. However using virtual particles cannot increase the order of the particle approximation accuracy in SPH. The virtual particles in FPM only act to implement the boundary conditions. Using virtual particles does not change the order of particle approximation accuracy in FPM. Different from the real particles, which can move or flow according to the force calculated from the particle–particle interactions, virtual particles are fixed in the problem space. The virtual particles contribute to the interactions of the flow particles, and they are also used to exert a repulsive boundary force to prevent the interior flow particles from penetrating the solid boundary. If a virtual particle is the neighboring particle of a flow particle that is approaching the boundary, a penalty force, which is usually taken the form similar to the Lennard-Jones molecular interaction [23,38], is applied pairwisely along the centerline of these two particles.

To model the effects of two infinite parallel plates, the periodic boundary condition was applied at the entrance and exit along the flow direction both in the particle movement and particle interaction. In the particle movement, a particle that leaves the specified region through a particular boundary face immediately reenters the region through the opposite face. In the particle interaction, a particle located within the dimension of the support domain from a boundary interacts with particles in an adjacent copy of the system, or equivalently with particles near the opposite boundary.

In the simulation, the problem domain was a rectangular of 0.0005 m × 0.001 m, and was modeled with 20 × 40 real particles of equal mass which were located at the geometric centers of a structured finite difference mesh (20 × 40) (Fig. 2). 2 × 20 virtual particles were deployed on the top and bottom solid boundaries, each with a mass equal to the mass of flow particles. Note the initial particle spacing between virtual particles and the outer layer of flow particles was only half of that between flow particles. The smoothing length used was 1.1 times the initial particle spacing. The time step was set to 10^{-4} s. After about 5000 steps (or 0.5 s), the flow reached a steady state. Figs. 3 and 4 show the comparison between the velocity profiles obtained using FPM, SPH and the analytical series solution [39] at $t = 0.01$ s, 0.1 s, and the final steady state $t = \infty$ for the two cases with $Re = 0.025$ and $Re = 0.1$. The FPM results agree very well with the analytical solutions, and the maximum error at the final steady state is less than 0.5%. The FPM results are much closer to the analytical solutions than the SPH results.

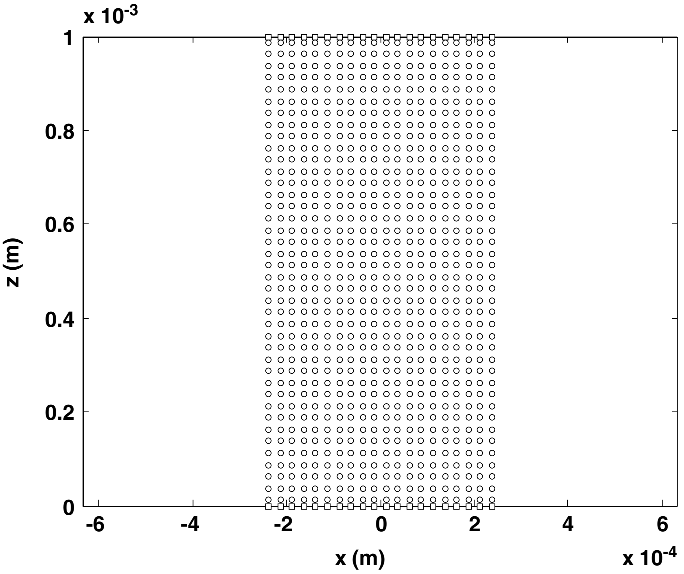

Fig. 2. Initial particle distributions for the Poiseuille flow and Couette flow. The circles represent real flow particles, whereas the squares represent the boundary virtual particles.

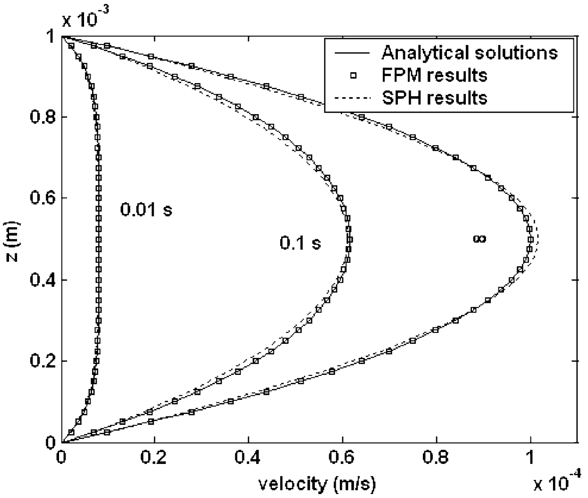

Fig. 3. Velocity profiles at different instants for the Poiseuille flow with a Reynolds number of $Re = 0.025$.

5.2. Couette flow

The Couette flow involves a fluid flow between two infinite plates that are initially stationary and placed at $y = 0$ and $y = l$. The flow is generated after the upper plate suddenly moves at a certain constant velocity $\boldsymbol{v}_0$ horizontally. In this example, $l = 10^{-3}$ m, $\upsilon = 10^{-6}$ m^2/s, $\rho = 10^3$ kg/m^3. Again

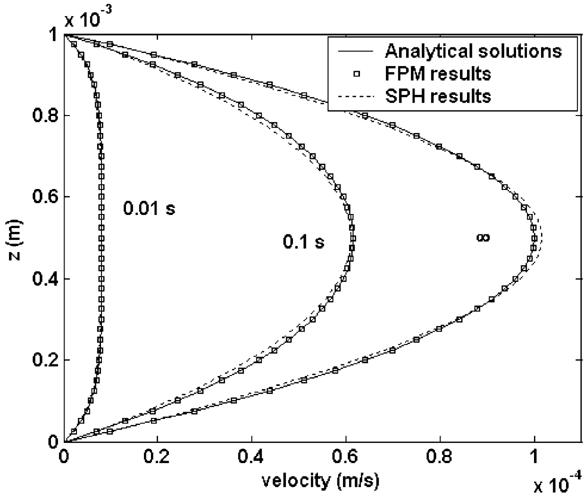

Fig. 4. Velocity profiles at different instants for the Poiseuille flow with a Reynolds number of $Re = 0.1$.

two cases were considered, one with $\boldsymbol{v}_0 = 2.5 \times 10^{-5}$ m/s^2, the other with $\boldsymbol{v}_0 = 1.0 \times 10^{-4}$ m/s^2, which correspond to Reynolds numbers of $Re = 0.025$ and $Re = 0.1$ respectively. The sound speed was taken as 2.5×10^{-4} and 1.0×10^{-3} m/s with $M = 0.1$. The geometry and the initial particle distribution were the same as the above example of Poiseuille flow (Fig. 2). The smoothing length and time step were also exactly the same as those in the above example. Similarly the flow reached a steady state after about 5000 steps (or 0.5 s). Figs. 5 and 6 show the comparison between the velocity profiles obtained using FPM, SPH and the analytical series solution [39] at $t = 0.01$ s, 0.1 s, and the final steady state $t = \infty$ for the two cases with $Re = 0.025$ and $Re = 0.1$. Again, the FPM results agree very well with the analytical solutions, and the maximum error at the final steady state is less than 0.5%. The FPM results are much closer to the analytical solutions than the SPH results.

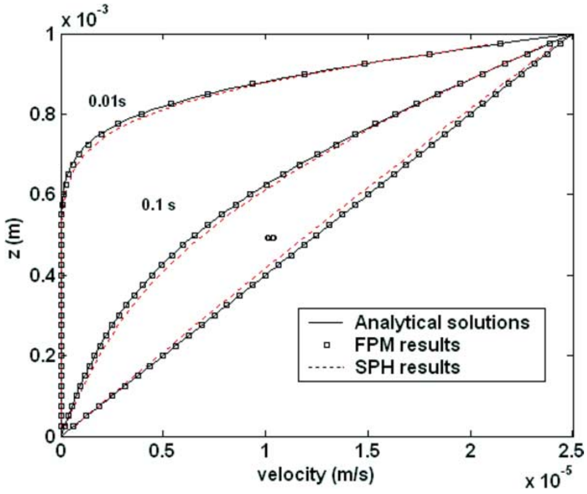

Fig. 5. Velocity profiles at different instants for the Couette flow with a Reynolds number of $Re = 0.025$.

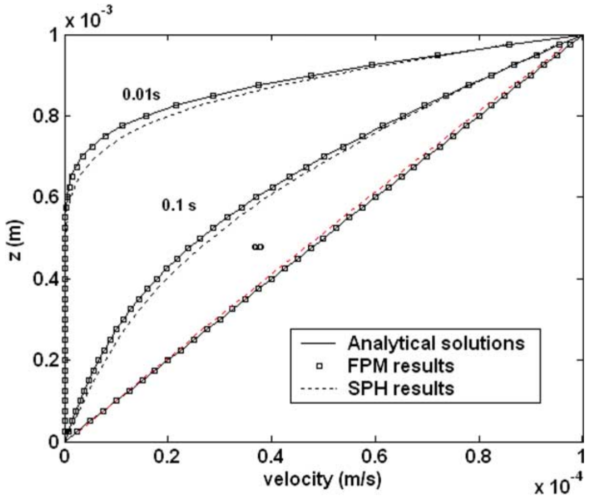

Fig. 6. Velocity profiles at different instants for the Couette flow with a Reynolds number of $Re = 0.1$.

5.3. Shear driven cavity

The classic shear driven cavity problem is the fluid flow within a closed square generated by horizontally moving the top side of the square at a constant velocity V_{Top} while the other three sides remain stationary. The flow will reach a steady state, and form a recirculation pattern. In the simulation, the dimension of the side of the square domain was $l = 10^{-3}$ m; the kinetic viscosity and density were $\upsilon = 10^{-6}$ m^2/s and $\rho = 10^3$ kg/m^3 respectively (for water). The top side of the square moved at a velocity of $V_{\mathrm{Top}} = 10^{-3}$ m/s, thus the Reynolds number for this case is was 1. A total of 1600 (40 × 40) real particles were evenly placed in the problem domain with another 320 (81 on each side of the square) boundary particles evenly located right on the four edges. Since the initially real particles were evenly distributed, the lumped volume for each real particle (m_i/ρ_i) were the square of the particle spacing. The particle spacing of the boundary particles were half of the particle spacing of real particles. The sound speed was taken as $0.1V_{\mathrm{Top}}$. A constant time step of 5×10^{-5} s was used. It takes approximately 3000 steps to reach a steady state.

Fig. 7 shows the non-dimensional vertical velocity profile along the horizontal centerline and the non-dimensional horizontal velocity profile along the vertical centerline respectively. The FDM (see [3]) results were obtained using a 40 × 40 grid, which is comparable to the FPM and SPH resolution of 40 × 40 real particles. It can be seen from Fig. 7 that the results obtained using FPM is very close to the FDM results. The results from SPH in some extent also agree with the FDM results. However, the accuracy of SPH in simulating this fluid flow problem is obviously lower than that of FPM.

5.4. Dam collapsing

Incompressible flows with free surfaces, moving interfaces and deformable boundaries are very important in industrial and environmental studies. As discussed in the introduction section, simulation of such flow problems using grid-based numerical methods usually requires special

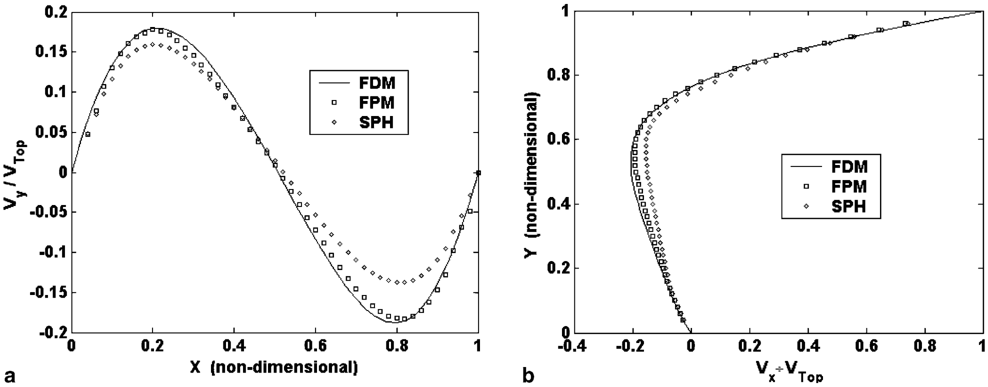

Fig. 7. Comparisons of the results obtained using FDM, SPH and FPM for the shear driven cavity problem: (a) vertical velocity profiles along the horizontal centerline at the steady state and (b) horizontal velocity profiles along the vertical centerline at the steady state.

techniques to track or capture the free surfaces, moving interfaces and deformable boundaries. One interesting incompressible flow problem with free surface is the dam collapsing problem which is frequently tested by new methods and commercial CFD code as a verification benchmark.

The FPM code was used to simulating a simplified, sudden dam collapsing problem [38] with an initial geometry of 25 m × 25 m with 50 × 50 uniformly distributed initial real particles and 216 virtual particles to model the solid walls and obstacle (Fig. 8).

Fig. 9 shows the particle evolutions at two representative instants obtained using FPM and SPH. The flow patterns obtained using SPH and FPM are basically similar while in SPH simulation, the surge front advances much faster than that in FPM simulation. The water particles flow orderly forward with increasing surge front *SR* and decreasing water lever *HT*. The numerical results from the present work (denoted by the subscript p), the experimental data (denoted by the subscript exp), and the results by Monaghan [38] (denoted by the subscript m) are compared and shown in Table 1. The surge front *SR* and the water lever *HT* at different instants are normalized by the initial water lever HT_0 the time is normalized by $\sqrt{HT_0/g}$, where *g* is the gravity constant. The FPM results, especially the surge fronts, are more accurate than what presented in [38] in which the SPH method was used. Table 2 shows the comparisons of computational effort using

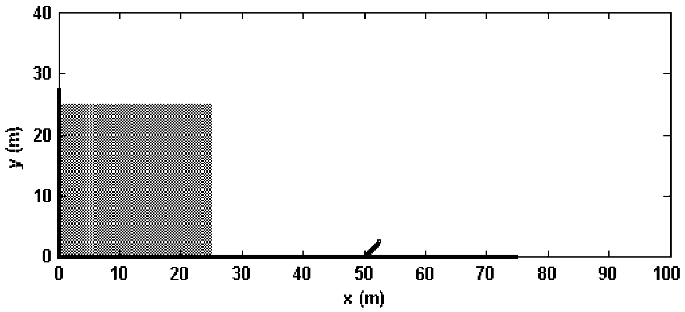

Fig. 8. Geometry and initial real and virtual particle distribution for the dam collapsing problem.

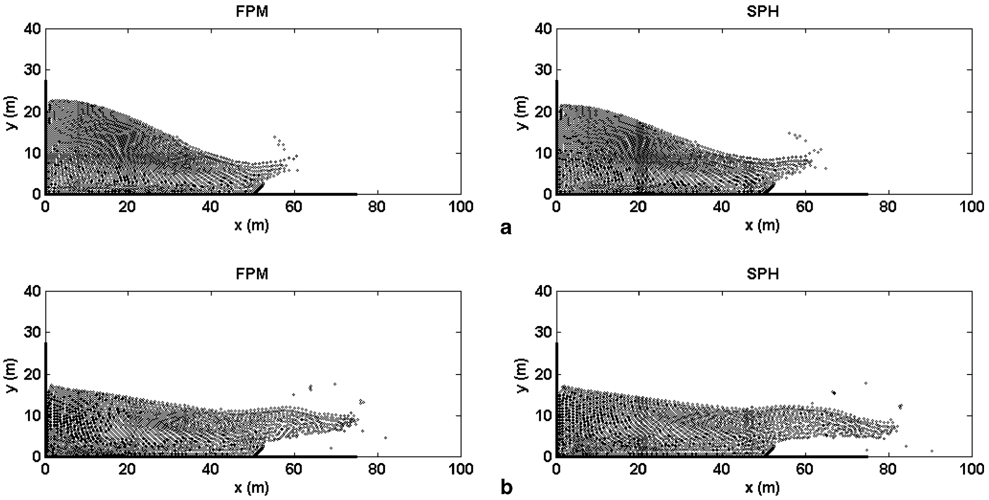

Fig. 9. Particle evolutions in the dam collapsing process at two typical stages: (a) $t/\sqrt{HT_0/g} = 1.45$ and (b) $t/\sqrt{HT_0/g} = 2.0$.

Table 1
Water level and water surging front for the dam collapsing problem

Time	HT_{exp}	HT_{m}	HT_{p}	SR_{exp}	SR_{m}	SR_{p}
0.71	0.90	0.90	0.90	1.33	1.56	1.41
1.39	0.76	0.75	0.75	2.25	2.50	2.35
2.10	0.57	0.56	0.56	3.22	3.75	3.45
3.20	0.32	0.37	0.34	3.80	5.00	3.86

Variables: *HT*—height of water; *SR*—surge front of water; Subscripts: p—present solution; m—results from Monaghan [38]; exp—experimental data [38].

Table 2
Comparisons of computational effort using SPH and FPM

Real time (s)	SPH CPU time (s)	FPM CPU time (s)
1.0	220.69	694.09
2.0	290.27	890.96
3.0	365.02	1249.3
4.0	433.38	1607.6
5.0	506.23	1804.8

SPH and FPM. The consumed CPU time for FPM simulation is more than (around 3–4 times) that for SPH simulation.

6. Conclusions

This paper presented the application of finite particle method (FPM) to simulating incompressible flow problems. In FPM, particles with lumped volumes are used to represent the state of a system. The particles either can be fixed in an Eulerian frame as interpolation points or can move in a Lagrangian frame as material particles. A set of basis functions are employed in FPM to approximate of a field variable and its derivatives through solving a pointwise matrix equation. Therefore FPM is a meshfree, Lagrangian and particle method with desired accuracy depending on the order of the derivatives retained in the Taylor series expansion process.

The finite particle method was then used to discretize the Navier–Stokes equation that governs fluid flows. The incompressible flows were modeled as slightly compressible flows via an equation of state. Four numerical examples including the classic Poiseuille flow, Couette flow, shear driven cavity and a dam collapsing problem were presented with comparisons to other sources. The FPM results were found to be in a good agreement with the exact results from other sources. The better performance of FPM in these simulations verified the previously discussed superiority of FPM to SPH in terms of approximation accuracy. It can be concluded that FPM is an attractive alternate for simulating incompressible flows with moving boundaries.

References

[1] T.J. Chung, Computational Fluid Dynamics, Cambridge University Press, New York, 2001.

[2] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York, 1980.

[3] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer, 2002.

[4] F.H. Harlow, The particle-in-cell method for numerical solution of problems in fluid dynamics, in: Proceedings of Symposia in Applied Mathematics, 1963.

[5] F.H. Harlow, E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluids with free surface, Phys. Fluids 8 (1965) 2182–2188.

[6] N.L. Johnson, The legacy and future of CFD at Los Alamos, in: Proceedings of the 1996 Canadian CFD Conference, Ottawa, Canada, 1996.

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

[8] C.W. Hirt, B.D. Nicholls, Volume of fluid (VOF) method for dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–221.

[9] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[10] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Month. Notices Roy. Astronom. Soc. 181 (1977) 375–389.

[11] L.B. Lucy, Numerical approach to testing the fission hypothesis, Astronom. J. 82 (1977) 1013–1024.

[12] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its applications in applied mechanics, Comput. Struct. 11 (1980) 83–95.

[13] E. Onate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, A finite point method in computational mechanics applications to convective transport and fluid flow, Int. J. Numer. Methods Engng. 39 (22) (1996) 3839–3866.

[14] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element methods: diffuse approximation and diffuse elements, Comput. Mech. 10 (1992) 307–318.

[15] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, Int. J. Numer. Methods Engng. 37 (1994) 229–256.

[16] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Int. J. Numer. Methods Fl. 20 (1995) 1081–1106.

[17] C.A. Duarte, J.T. Oden, An HP adaptive method using clouds, Comput. Methods Appl. Mech. Engng. 139 (1996) 237–262.

[18] G. Yagawa, T. Yamada, Free mesh method: a new meshless finite element method, Comput. Mech. 18 (1996) 383–386.

[19] S.N. Atluri, T. Zhu, A new meshless local Petrov–Galerkin (MPLG) approach in computational mechanics, Comput. Mech. 22 (1998) 117–127.

[20] G.R. Liu, Y.T. Gu, A point interpolation method for two-dimensional solids, Int. J. Numer. Methods Engng. 50 (2001) 937–951.

[21] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recently developments, Comput. Methods Appl. Mech. Engng. 139 (1996) 3–47.

[22] S.F. Li, W.K. Liu, Meshfree and particle methods and their applications, Appl. Mech. Rev. 55 (1) (2002) 1–34.

[23] G.R. Liu, M.B. Liu, Smoothed Particle Hydrodynamics—A Meshfree Particle Method, World Scientific, 2003.

[24] Century Dynamics Incorporated, AUTODYN Release Notes Version 3.1, AUTODYN^{TM} Interactive Non-Linear Dynamic Analysis Software, 1997.

[25] J.J. Monaghan, Smoothed particle hydrodynamics, Ann. Rev. Astronom. Astrophys. 30 (1992) 543–574.

[26] L. Hernquist, N. Katz, TreeSPH—a unification of SPH with the hierarchical tree method, Astrophys. J. Suppl. Ser. 70 (1989) 419–446.

[27] G.R. Johnson, R.A. Stryk, S.R. Beissel, SPH for high velocity impact computations, Comput. Methods Appl. Mech. Engng. 139 (1996) 347–373.

[28] P.W. Randles, L.D. Libersky, Smoothed particle hydrodynamics some recent improvements and applications, Comput. Methods Appl. Mech. Engng. 138 (1996) 375–408.

[29] M.B. Liu, G.R. Liu, K.Y. Lam, Investigations into water mitigations using a meshless particle method, Shock Waves 12 (3) (2002) 181–195.

[30] M.B. Liu, G.R. Liu, K.Y. Lam, Z. Zong, Computer simulation of the high explosive explosion using SPH methodology, Comput. Fluids 32 (3) (2003) 305–322.

[31] M.B. Liu, G.R. Liu, Z. Zong, K.Y. Lam, Smoothed particle hydrodynamics for numerical simulation of underwater explosions, Comput. Mech. 30 (2) (2003) 106–118.

[32] M.B. Liu, G.R. Liu, Restoring particle consistency in smoothed particle hydrodynamics, Appl. Numer. Math., in press.

[33] J. Bonet, S. Kulasegaram, Correction and stabilization of smoothed particle hydrodynamics method with applications in metal forming simulations, Int. J. Numer. Methods Engng. 47 (2000) 1189–1214.

[34] G.A. Dilts, Moving least square particle hydrodynamics i: consistency and stability, Int. J. Numer. Methods Engng. 44 (1999) 1115–1155.

[35] J.K. Chen, J.E. Beraun, A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems, Comput. Methods Appl. Mech. Engng. 190 (2000) 225–239.

[36] M.B. Liu, G.R. Liu, K.Y. Lam, Constructing smoothing functions in smoothed particle hydrodynamics with applications, J. Comput. Appl. Math. 155 (2) (2003) 263–284.

[37] M.B. Liu, G.R. Liu, On the selection of basis function in finite particle method, Comput. Mech., in press.

[38] J.J. Monaghan, Simulating free surface flow with SPH, J. Comput. Phys. 110 (1994) 399–406.

[39] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1997) 214–226.

[40] G.K. Batchelor, An Introduction to Fluid Dynamics, Cambridge University Press, Cambridge, UK, 1973.

[41] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.

[42] S.E. Rogers, D. Kwak, U.K. Kaul, On the accuracy of the pseudocompressibility method in solving the incompressible Navier–Stokes equations, Appl. Math. Model. 11 (1987) 35–44.

[43] G.M. Zhang, R.C. Batra, Modified smoothed particle hydrodynamics method and its applications to transient problems, Comput. Mech. 34 (2004) 137–146.

[44] R.C. Batra, G.M. Zhang, Analysis of adiabatic shear bands in elasto-thermo-viscoplastic materials by modified smoothed-particle hydrodynamics (MSPH) method, J. Comp. Phys. 201 (2004) 172–190.