Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 325 (2017) 77–101
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

High-order particle method for solving incompressible Navier–Stokes equations within a mixed Lagrangian–Eulerian framework

Kuan-Shuo Liu[a], Tony Wen-Hann Sheu[a,b,*], Yao-Hsin Hwang[c], Khai-Ching Ng[d,*]

[a] *Department of Engineering Science and Ocean Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan*
[b] *Center for Advanced Study in Theoretical Sciences, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan*
[c] *Department of Marine Engineering, National Kaohsiung Marine University, Kaohsiung 805, Taiwan*
[d] *Center of Fluid Dynamics (CFD), Department of Mechanical Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia*

Received 27 February 2017; received in revised form 2 June 2017; accepted 3 July 2017
Available online 14 July 2017

Highlights

- A mixed-Lagrangian–Eulerian (MLE) method is proposed.
- The total derivative terms are solved in the Lagrangian sense.
- The spatial derivative terms are solved in the Eulerian.
- The continuity equation is solved on Cartesian mesh to retain the elliptic nature.
- Data transfer between particles and grids are realized by interpolations.

Abstract

Owing to the fact that the Poisson equation of pressure for incompressible fluid flow is purely elliptic, it is therefore computationally improper to compute pressure on irregularly distributed moving particles as addressed in the recent Moving Particle with embedded Pressure Mesh (MPPM) method. In the current work, a modified MPPM method known as the Mixed Lagrangian–Eulerian (MLE) method is proposed for solving the incompressible Navier–Stokes equations. In the current velocity–pressure formulation, the momentum and continuity equations are approximated on the moving particles (Lagrangian) and the uniform Cartesian grid points, respectively. Meanwhile, the total derivative of velocity terms appeared in the momentum equations are estimated by simply advecting the moving particles, thereby eliminating the convection stability problem and increasing the flow accuracy without introducing false diffusion error. In the conventional Moving Particle Semi-implicit (MPS) and MPPM methods, numerical accuracies of the Laplacian and gradient operators are strongly dependent on the regularity of the particle distribution. In some implicit schemes, the gradient and Laplacian terms are of second-order and first-order accuracy, respectively. In the current work, the second-order accuracies of these differential terms exhibited on moving particles are realized by interpolating the derivative values from the uniform Cartesian grids calculated by using the high-order Combined Compact Difference (CCD) scheme. From the numerical results of Laplacian term approximation by using various numerical schemes, it is shown that the new

* Corresponding authors.
E-mail addresses: f01525002@ntu.edu.tw (K.-S. Liu), twhsheu@ntu.edu.tw (T.W.-H. Sheu), yhhwang@webmail.nkmu.edu.tw (Y.-H. Hwang), ngkhaiching2000@yahoo.com (K.-C. Ng).

http://dx.doi.org/10.1016/j.cma.2017.07.001

MLE scheme is at least second-order accurate. The proposed Mixed Lagrangian–Eulerian (MLE) method can be easily applied to simulate fluid flow problems ranging from low to high Reynolds number. It is found that the numerical results compare well with the benchmark solutions. Moreover, it is more accurate than the recently proposed MPPM method.

Keywords: Moving particle Semi-implicit; Moving particle with embedded pressure mesh; Mixed Lagrangian–Eulerian

1. Introduction

Numerical methods developed to solve partial differential equations can be broadly divided into grid-based and meshless categories. In grid-based methods, the conventional Finite Difference Method (FDM), Finite Volume Method (FVM) and Finite Element Method (FEM) rely on differential operators underlying the respective Taylor series expansion at all interior points, conservation of equations within each control volume, and the variational principle over the entire physical domain, respectively. Moreover, the respective rigorous mathematical methodologies developed to analyze the corresponding discretization errors exist. In contrast to mesh-based methods, meshfree methods do not require mesh connectivity in the simulation domain. Meshfree methods, as a result, are more effective in simulating some complex manufacturing processes involving large interface deformation such as extrusion and molding as compared to mesh-based methods. In this study, our focus is to combine the meshfree and the mesh-based methods to predict the incompressible Navier–Stokes equations.

A genuine meshfree method dates back to 1965 [1]. Later on, in 1977, Gingold and Monaghan [2] and Lucy [3] developed a Lagrangian method known as the Smoothed Particle Hydrodynamics (SPH) method used to model astrophysics problems. In SPH, mesh points are replaced by a group of moving particles. The numerical approximation in SPH is performed in weak (integral) form. Despite its success in early days, SPH method suffers from numerical stability due to the lack of the required interpolation consistency and the proper implementation of essential or natural boundary condition [4]. These computational difficulties have motivated many researchers to work on methods used to restore the consistency property [5] and to resolve the difficulty of enforcing essential boundary condition [4]. The explicit type SPH method which is suitable for weakly-compressible flow simulation has now been successfully applied to solve a wide range of engineering problems as detailed in [6,5].

The Moving Particle Semi-implicit (MPS) method is another class of meshfree methods developed by Koshizuka and Oka [7] in 1996. Similar to SPH, MPS method approximates the strong form of the partial differential equations by a simple local weighted averaging without the need of expressing the kernel function in differential form. MPS method differs from the fully explicit SPH method in that MPS requires a semi-implicit prediction–correction iterative procedure. Also, MPS has a prevailing potential in solving incompressible flow equations while the SPH method is more suitable for solving the compressible flow equations.

MPS method has been successfully developed to simulate incompressible free surface flow without the necessity of tracking the free surface by scalar quantities such as the Volume of Fluid (VOF) and the level-set function. While MPS method can be used to predict free surface flow involving either complex coalescence or breakup in a relatively easy manner (i.e. without interface tracking), the pressure field suffers from numerical stability nevertheless. Also, it is computationally expensive to solve the Poisson equation of pressure on moving particles [8] because the coefficient matrix must be correspondingly modified due to the change of positions of particles after each time step. Moreover, the divergence-free continuity equation, i.e. $\nabla \cdot \vec{u} = 0$, is not directly solved, and it may lead to the wiggling pressure field. Several improved versions of the MPS method addressing on the enhancement of numerical stability have been proposed [9–11,8,12,13].

For the purpose of resolving the problem of pressure oscillations, Monaghan [14] proposed a completely different SPH method by solving the weakly compressible Navier–Stokes equations rather than the incompressible one as outlined in the MPS method. Pressure smoothing is achieved via introducing the fluid density term, which is considered as an extra unknown in the SPH formulations. It is noted that SPH method has been extended to predict incompressible free surface flow [15–17]. One can refer to the work of Shao and Gotoh [18] for the comparison of incompressible SPH and MPS methods.

Other gridless algorithms include the least-square method [19], the CIP (Cubic Interpolated Pseudo particle) scheme [20], and the MAFL (Meshless Advection using Flow-direction Local grid) method [21]. For a review of other

meshless methods such as Diffusive Element Method (DEM), Finite Point Method, Meshless Local Petrov–Galerkin method (MLPG), radial basis functions method, and the local boundary integral method, one can refer to [22–27].

In fact, for almost all the particle-based methods mentioned above, the Laplacian operators (viscous stress term) are discretized on the particle level. Due to the irregularity of particle topology which is inevitable, one has to employ more neighboring particles (enlargement of radius of influence) in order to attain a numerical approximation of higher-order accuracy. This is rather costly from the computational point of view. In the recently proposed MPPM method, a Cartesian background mesh has been employed to discretize the continuity equation.

In [28], Hwang proposed the MPPM method that solves the PPE on the Cartesian mesh to suppress pressure oscillations. Mass conservation is guaranteed at the Eulerian mesh level and moving particles are simply viewed as observation points which can be added or deleted during the simulation. For transport equation such as momentum equations, the local and convective accelerations are combined (hence total acceleration) in order to get rid of the convective instability problem normally encountered in Eulerian methods. The pressure gradients of moving particles are calculated from pressure values stored in the Eulerian mesh. Meanwhile, the Laplacian terms are discretized in the Lagrangian fashion similar to the conventional particle methods. Several improvements on MPPM have been recently proposed. For example, Hwang has proposed Local Mesh (LM) [29] and Improved Particle Smoothing (IPS) schemes to evaluate the Laplacian terms of moving particles. Ng et al. [30] have simulated multi-phase flows with MPPM combined with the conservative level-set method. More recently, Ng et al. [31] have proposed UMPPM method to deal with flows involving complex geometry, showing the potential of MPPM-based methods to simulate practical flow problems.

In the current work, we intend to improve the accuracy of MPPM method. In essence, the calculation of the Laplacian term in the current work is divided into two parts. We find that the Laplacian term on the background mesh (instead of the particle level) can be approximated by using a higher-order accurate compact scheme. Subsequently, these approximations are interpolated to the particle level. In the current method, the total derivative terms are calculated on Lagrangian moving particles to avoid convective instability (similar to conventional particle method). On the other hand, spatial derivative terms such as pressure gradient, velocity Laplacian and velocity divergence are calculated from the pressure and velocity values stored in the Eulerian mesh in order to attain higher accuracy. Our method is called the Mixed Lagrangian–Eulerian (MLE) method in the current paper.

This paper is organized as follows. In Section 2, the equations for modeling the incompressible fluid flow are presented in their primitive-variable forms. The solutions of this elliptic–parabolic system of equations subjected to the theoretically rigorous velocity boundary condition are sought. Section 3 describes the proposed Mixed Lagrangian–Eulerian (MLE) semi-implicit solution algorithm. Section 4 is dedicated to the computations of momentum equations and pressure Poisson equation in Lagrangian (moving particle) and Eulerian (Cartesian grids) descriptions, respectively. The key to success in the proposed MLE formulation lies on the employed scheme used for the interpolations of $\vec{u}$ (flow velocity vector) and p (pressure) between the moving particle positions and the fixed Eulerian nodes. Section 5 is aimed to verify the developed code by solving the problems amenable to exact solutions. Also, some benchmark problems are investigated in Section 6 to validate the MLE formulation and the computer program. Finally, we will conclude our findings in Section 7.

2. Semi-discretization of the incompressible Navier–Stokes equations

In this paper, the following elliptic–parabolic equations for the primitive variables $\vec{u}$ (velocity) and p (pressure) are considered:

$$\nabla \cdot \vec{u} = 0 \tag{1}$$

$$\frac{D\vec{u}}{Dt} \equiv \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{u}. \tag{2}$$

In the above primitive-variable formulation, ρ and ν denote the fluid density and the kinematic viscosity, respectively. The reason for choosing the above set of equations to simulate the hydrodynamic phenomena of incompressible fluid flow in a complex domain Ω is due to the existence of theoretical specification of velocity on the boundary $\partial\Omega$. Another reason is that the variables $\vec{u}$ and p are more easily understood. The solution of this mixed type elliptic–parabolic differential equations is sought subjected to the initial condition $\vec{u}(\vec{x}, t = 0) = \vec{u}_0$.

For the above set of equations, the spatial derivative terms such as the pressure gradient and the velocity Laplacian terms will be calculated in Eulerian sense to improve the solution accuracy. Meanwhile, the unknown p will be calculated from the Pressure Poisson equation (PPE) on Cartesian mesh to retain the nature of ellipticity. On the other hand, the total derivative $D\vec{u}/Dt$ shown in the left hand side of Eq. (2) will be solved in Lagrangian sense by using the moving particle method. The reason behind this strategy is to get rid of the convective instability problem and to reduce the indispensable false diffusion error in grid-based methods.

To solve the primitive-variable Navier–Stokes equations, the projection method is adopted. In the first step, momentum equations without pressure gradient term are solved explicitly to get the intermediate solutions. Meanwhile, particle location is updated as well according to:

$$\vec{u}_P^* = \vec{u}_P^n + \Delta t \nu \nabla^2 \vec{u}_P^n \tag{3}$$

$$\vec{r}_P^* = \vec{r}_P^n + \Delta t \vec{u}_P^*. \tag{4}$$

At the current stage, the velocity field does not satisfy the divergence-free condition. A modified step is required to get a divergence-free velocity field:

$$\vec{u}_P^{n+1} = \vec{u}_P^* - \frac{\Delta t}{\rho} \nabla p_P^{n+1} \tag{5}$$

$$\vec{r}_P^{n+1} = \vec{r}_P^* - \frac{\Delta t^2}{\rho} \nabla p_P^{n+1}. \tag{6}$$

Since the velocity field shown in Eq. (5) should be divergence-free, a pressure Poisson equation is derived by substituting (5) into Eq. (1). Therefore, we can get

$$\nabla^2 p_G^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \vec{u}_G^*. \tag{7}$$

In the above projection method, the superscripts n, $*$, and $n+1$ denote current, intermediate, and new time steps, respectively. The subscripts P and G denote the positions of moving particles and stationary grid points, respectively.

3. Mixed Lagrangian–Eulerian (MLE) method

Mesh-based methods have been applied with great success in the exploration of complex physics in many industrial applications. In this class of methods, distorted meshes can lead to larger numerical errors. Hence, a costly remeshing process is normally required. The pre-processing effort required while simulating fluid flow in three-dimensional complex geometries is enormous. Moreover, mesh-based methods are not suitable for handling problems with discontinuities that do not align with the mesh lines. Meshless methods, on the other hand, approximate the spatial derivative terms on the nodal basis. Meshless methods can be developed in strong form (e.g. MPS), weak form (e.g. element-free Galerkin method), and local weak form (e.g. local Petrov–Galerkin method). While meshless methods seem to be able to tackle large deformation problems more naturally, they are comparatively cumbersome to be implemented and are computationally more expensive. Numerical methods such as the extended finite element method [32] have been developed by combining the advantages of mesh-based and meshless methods.

Moving Particle (MP) method is, in essence, one of the fully Lagrangian methods. MP method is therefore fundamentally suitable for solving transport equations for an incompressible fluid flow, particular for that of high Reynolds number.

3.1. Some existing schemes

To calculate the intermediate velocity components and the particle locations, the velocity Laplacian term $\nabla^2 \vec{u}_P^n$ in (3) must be firstly obtained. In the past few years, several Lagrangian schemes have been developed to approximate

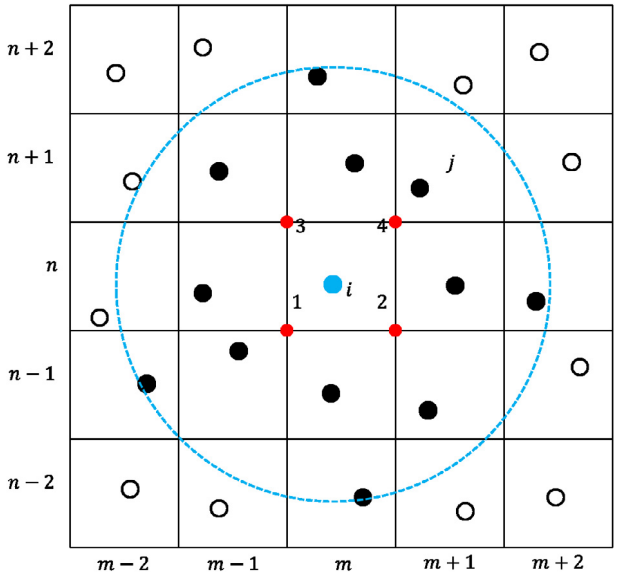


Fig. 1. Schematic of the old influence domain denoted as ID1 (Influence Domain 1) for particle i. Only the solid black circles are involved in the numerical operation.

the first and second order derivative terms in a mesh-free manner. However, the order of accuracy is ranged from $O(h^{-1})$ to $O(h^1)$. In our MLE method, a second-order accurate scheme is proposed.

Four existing Lagrangian schemes used for the discretization of Laplacian operator will be examined:

(1) Improved MPS scheme (MP) [33],

$$\nabla^2\phi_i = \frac{2d}{\sum_{j\neq i}\omega_{ij}}\sum\frac{\omega_{ij}}{r_{ij}^2}\phi_{ij}. \tag{8}$$

Here, i is the local particle, j is the neighboring particle within the influence domain of particle i as shown in the schematic diagram (see Fig. 1), d is the dimension of the problem, ω_{ij} is the weighting function, r_{ij} is the distance between particles i and j, and $\phi_{ij} = \phi_j - \phi_i$. The weighting function used in this research work takes the form of [34].

$$\omega(r) = \begin{cases} \dfrac{a}{r_e} + \dfrac{b}{r_e}\left(\dfrac{r}{r_e}\right) + \dfrac{c}{r_e}\left(\dfrac{r}{r_e}\right)^2 + \dfrac{d}{r_e}\left(\dfrac{r}{r_e}\right)^3 + \dfrac{e}{r_e}\left(\dfrac{r}{r_e}\right)^4 & 0 \le r \le r_e \\ 0 & r_e > r \end{cases} \tag{9}$$

where

$$a = \frac{480\sqrt{2}-705}{512\sqrt{2}-745} \quad b = 0 \quad c = \frac{-960\sqrt{2}+1515}{512\sqrt{2}-745}$$

$$d = \frac{-210}{512\sqrt{2}-745} \quad e = \frac{480\sqrt{2}-600}{512\sqrt{2}-745}.$$

(2) Local Mesh (LM) method [35],

$$\nabla^2\phi_i = (\nabla\phi_{en}\cdot\vec{A}_{en} + \nabla\phi_{nw}\cdot\vec{A}_{nw} + \nabla\phi_{ws}\cdot\vec{A}_{ws} + \nabla\phi_{se}\cdot\vec{A}_{se})/V_{enws} \tag{10}$$

where $\vec{A}_{ij} = (y_j - y_i)\vec{i} + (x_j - x_i)\vec{j}$. It is worth noting that the LM scheme can be only used for interior particles, i.e. particles in cell $(2, 2) \sim (n-1, n-1)$ if there are n cells along x and y -direction. For particles in boundary cells, some other schemes should be used.

(3) Smoothing Difference (SD) method [29],

$$Cq = b. \tag{11}$$

In the above, $q = [\phi_x \quad \phi_y \quad \phi_{xx} \quad \phi_{xy} \quad \phi_{yy}]^T$. The coefficient matrix C and the source term b shown in (11) are detailed in [29].

(4) Generalized Finite Difference (GFD) method [36],

$$\nabla^2\phi_i = \sum_{j\neq i}\omega_{ij}^2\left[(c_1+e_1)h_{ij}+(c_2+e_2)k_{ij}+(c_3+e_3)\frac{h_{ij}^2}{2}+(c_4+e_4)h_{ij}k_{ij}+(c_5+e_5)\frac{k_{ij}^2}{2}\right]. \tag{12}$$

The GFD method is derived based on the theory of Taylor series expansion. For a second-order Taylor series expansion, the five derivative terms shown in q are obtained by solving a linear system with at least five particles inside the influence domain of particle i. However, the accuracy can be improved by adopting more than five particles. The coefficient matrix of the linear system can be obtained by using the least squares method through the minimization of the square of errors. The coefficient matrix can be derived by following the work of Koh et al. [37],

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5\\ b_1 & b_2 & b_3 & b_4 & b_5\\ c_1 & c_2 & c_3 & c_4 & c_5\\ d_1 & d_2 & d_3 & d_4 & d_5\\ e_1 & e_2 & e_3 & e_4 & e_5 \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j\neq i}\omega_{ij}^2h_{ij}^2 & \sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij} & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}^3 & \sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij} & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij}^2\\ \sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij} & \sum_{j\neq i}\omega_{ij}^2k_{ij}^2 & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij} & \sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij}^2 & 0.5\sum_{j\neq i}\omega_{ij}^2k_{ij}^3\\ 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}^3 & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij} & 0.25\sum_{j\neq i}\omega_{ij}^2h_{ij}^4 & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}^3k_{ij} & 0.25\sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij}^2\\ \sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij} & \sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij}^2 & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}^3k_{ij} & \sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij}^2 & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij}^3\\ 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij}^2 & 0.5\sum_{j\neq i}\omega_{ij}^2k_{ij}^3 & 0.25\sum_{j\neq i}\omega_{ij}^2h_{ij}^2k_{ij}^2 & 0.5\sum_{j\neq i}\omega_{ij}^2h_{ij}k_{ij}^3 & 0.25\sum_{j\neq i}\omega_{ij}^2k_{ij}^4 \end{bmatrix}^{-1} \tag{13}$$

However, if GFD method is used, a local 5 × 5 matrix must be constructed and inverted at every time step for each particle in order to compute the Laplacian operator. It is of course very time consuming if the number of particles is large. A more detailed study on the accuracy order and the computational time (CPU time) required will be shown in the next section.

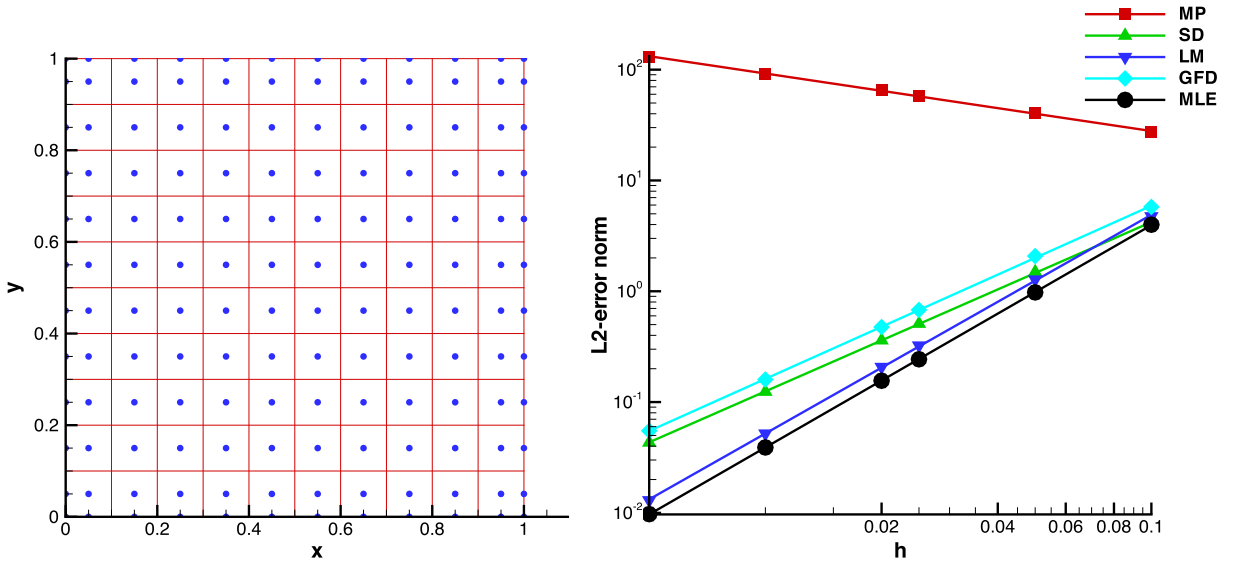
3.2. The proposed second-order accurate Laplacian operator on moving particles

It is well accepted that particle distribution strongly affects the accuracy order of the spatial derivative schemes. It is therefore obvious that better accuracy can be obtained on uniform Cartesian mesh. To improve the accuracy order of the Laplacian operator shown in the momentum equations, the derivative term ($\nabla^2\vec{u}_G^n$) is firstly calculated on Cartesian mesh using the high-order Combined Compact Difference (CCD) scheme proposed by Chu and Fan [38]. The following equations are solved to get the sixth-order accurate solutions of the first and second order derivative terms simultaneously.

$$\frac{7}{16}\left(\left(\frac{\delta f}{\delta x}\right)_{i+1}+\left(\frac{\delta f}{\delta x}\right)_{i-1}\right)+\left(\frac{\delta f}{\delta x}\right)_i-\frac{h}{16}\left(\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1}-\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}\right)=\frac{15}{16h}\left(f_{i+1}-f_{i-1}\right) \tag{14}$$

$$\frac{9}{8h}\left(\left(\frac{\delta f}{\delta x}\right)_{i+1}-\left(\frac{\delta f}{\delta x}\right)_{i-1}\right)-\frac{1}{8}\left(\left(\frac{\delta^2 f}{\delta x^2}\right)_{i+1}+\left(\frac{\delta^2 f}{\delta x^2}\right)_{i-1}\right)+\left(\frac{\delta^2 f}{\delta x^2}\right)_i=\frac{3}{h^2}\left(f_{i+1}-2f_i+f_{i-1}\right). \tag{15}$$

The current MLE method is featured with the calculation of spatial derivative terms on the Eulerian mesh and the results are then interpolated to the Lagrangian particles. A simple second-order interpolation scheme is used to get



(a) Particle distribution.

(b) Rate of convergence.

Fig. 2. Test of Laplacian operator for the case of $\alpha = 0.0$.

a globally second-order accurate Laplacian approximation. For the two-dimensional case, the bilinear interpolation function is chosen as

$$\nabla^2\phi_P = \sum_{i=1}^{4} N_i(\nabla^2\phi_G)_i \tag{16}$$

where N_i are the shape functions at the four corner red points (see Fig. 1) surrounding the local particle i. The shape functions N_i corresponding to the four corner points are:

$$N_1 = (1-\xi)(1-\eta) \qquad N_2 = \xi(1-\eta)$$
$$N_3 = (1-\xi)\eta \qquad N_4 = \xi\eta$$

where

$$\xi = \frac{x_i - x_1}{\Delta x} \qquad \eta = \frac{y_i - y_1}{\Delta y}.$$

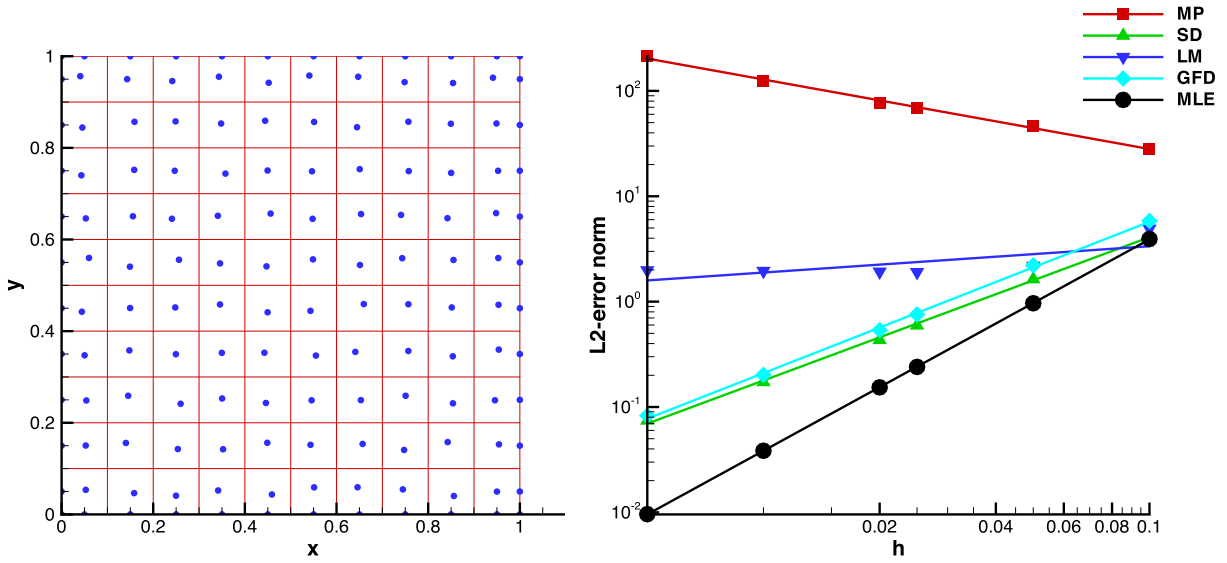
In order to compute the orders of accuracy of several Laplacian schemes, a test function $f(x, y) = \sin(2\pi x)\sin(2\pi y)$ is used. Here, we are interested to calculate the values of $\nabla^2 f$ on each particle and the predicted values are compared with the analytical solution: $\nabla^2 f(x, y) = -8\pi^2 \sin(2\pi x)\sin(2\pi y)$. Here, the effect of particle distribution is also examined. Particles are seeded in each cell by the following equations

$$x = x_c + \alpha\beta \cdot \Delta x \tag{17}$$

$$y = y_c + \alpha\beta \cdot \Delta y. \tag{18}$$

In the above, x and y denote the particle locations, x_c and y_c are the cell-center coordinates, α is the prescribed regularity parameter ($0 \le \alpha \le 1$) and β is a uniform random number lying between -0.5 and 0.5. The condition of $\alpha > 0$ is enforced in order to randomize particle locations. We choose $\alpha = 0.0, 0.2, 0.8$ to study the effect of regularity on the accuracy order of the predicted solution. The particle distribution and the rate of convergence are shown in Figs. 2–4.

For the spatial derivative terms, the orders of accuracy of several existing Lagrangian schemes are summarized in Table 1. As expected, the newly proposed MLE method exhibits second order accuracy regardless of α. Also,



(a) Particle distribution. (b) Rate of convergence.

Fig. 3. Test of Laplacian operator for the case of $\alpha = 0.2$.

Table 1
Comparison of several Laplacian operator schemes for the case of $\alpha = 0.8$.

Scheme	MP	LM	SD	GFD	MLE
R.O.C.	−0.9141	−0.0490	1.0189	1.0572	2.0124

the required CPU time is listed in Table 2. It is interesting to note that our MLE method is more accurate and computationally more efficient than other methods.

It is important to note that the MP scheme is second-order accurate only for uniformly distributed interior particles. By applying Taylor series expansion to Eq. (8), we obtain

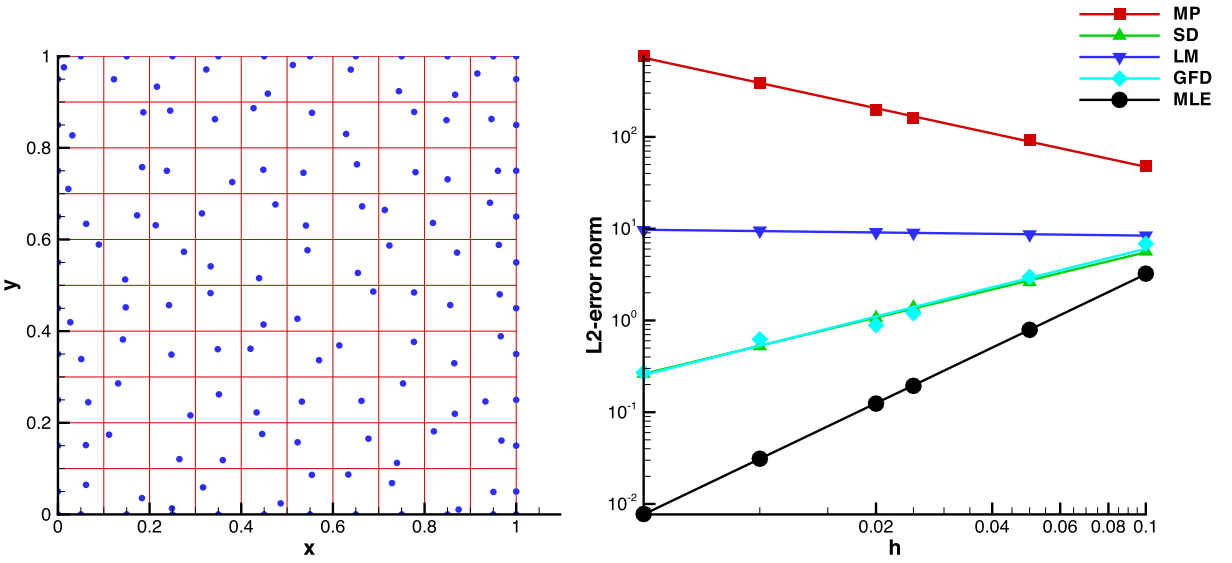
$$\frac{4}{\sum_{q\neq p}\omega_{qp}}\sum_{q\neq p}\frac{(\phi_q-\phi_p)\,\omega_{qp}}{r_{qp}^2}=\frac{4}{\sum_{q\neq p}\omega_{qp}}\left[\phi_x\sum_{q\neq p}\frac{(x_q-x_p)\,\omega_{qp}}{r_{qp}^2}+\phi_y\sum_{q\neq p}\frac{(y_q-y_p)\,\omega_{qp}}{r_{qp}^2}\right.$$
$$\left.+\frac{1}{2}\phi_{xx}\sum_{q\neq p}\frac{(x_q-x_p)^2\omega_{qp}}{r_{qp}^2}+\phi_{xy}\sum_{q\neq p}\frac{(x_q-x_p)\,(y_q-y_p)\,\omega_{qp}}{r_{qp}^2}+\frac{1}{2}\phi_{yy}\sum_{q\neq p}\frac{(y_q-y_p)^2\omega_{qp}}{r_{qp}^2}\right]+H.O.T.$$

where $H.O.T.$ stands for High Order Terms. It is obvious that the leading error is of order $O(1/\delta)$; therefore, NP scheme is numerically inconsistent (as witnessed in Figs. 2–4). In order to recover the second-order accurate Laplacian scheme, the following conditions must be satisfied:

$$\sum_{q\neq p}\frac{(x_q-x_p)\,\omega_{qp}}{r_{qp}^2}=0\quad\sum_{q\neq p}\frac{(y_q-y_p)\,\omega_{qp}}{r_{qp}^2}=0\quad\sum_{q\neq p}\frac{(x_q-x_p)\,(y_q-y_p)\,\omega_{qp}}{r_{qp}^2}=0$$

$$\sum_{q\neq p}\frac{(x_q-x_p)^2\omega_{qp}}{r_{qp}^2}=\sum_{q\neq p}\frac{(y_q-y_p)^2\omega_{qp}}{r_{qp}^2}=\frac{1}{2}\sum_{q\neq p}\omega_{qp}.$$

For uniformly distributed particles ($\alpha = 0$), only interior particles satisfy the above conditions. Those particles lying near boundaries do not fulfill the above conditions due to the truncation of particles beyond the problem domain. The



(a) Particle distribution.

(b) Rate of convergence.

Fig. 4. Test of Laplacian operator for the case of $\alpha = 0.8$.

Table 2
Comparison of the CPU times with respect to that of the MLE method for the case with 1000^2 particles.

	MPS	SD	LM	GFD	MLE
CPU time (s)	1.0316	1.4421	0.9816	1.1711	1.0000

Table 3
Rates of convergence for MP scheme applied solely in interior particles for the case of $\alpha = 0.0$.

Grid no.	L_2-error norm	R.O.C.
11	2.3943E+000	–
21	1.0151E+000	1.54
41	2.4224E−001	2.05
81	5.6825E−002	2.06
161	1.3694E−002	2.04
321	3.3596E−003	2.02

ROCs for MP scheme for interior particles are shown in Table 3. It indicates that MP scheme is second-order accurate for interior particles. However, MP scheme is numerically inconsistent as reported in Figs. 2–4 due to the boundary effect. Meanwhile, our MLE method provides a global second-order accuracy regardless for interior or near-boundary particles. Also, the computational time of MLE method is relatively lower than those of the existing methods as shown in Table 2.

3.3. New influence domain and particle chosen strategy

The common challenge in most of the particle-based methods lies in how to choose the neighboring particles or data points for differential operator calculation or interpolation [28]. For example, in order to identify the neighboring particles which are residing within the radius of influence ($r_e = \kappa h$) of particle i (located in cell (m, n)), particle information within κ cells away from the cell (m, n) (or cells from $(m - \kappa, n - \kappa)$ to $(m + \kappa, n + \kappa)$) are processed. However, as shown in Fig. 1 ($\kappa = 2$), particle in cell $(m - 2, n - 1)$ is included in the numerical operation of particle

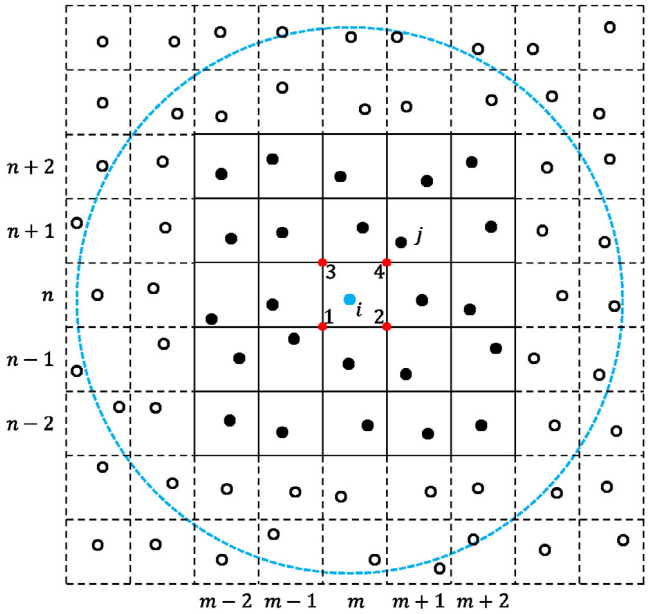


Fig. 5. Schematic of the new influence domain ID2 (Influence Domain 2) for particle i. Only the solid black circles are involved in the numerical operation.

i. Particles residing in cells $(m+2, n-1)$, $(m-1, n-2)$, $(m-2, n+1)$, $(m-2, n)$, and $(m+2, n+1)$, however, are not included. This condition would somehow cause the non-uniformity of particle distribution.

To overcome this problem, we propose a new method to calculate the radius of influence. If we wish to include all particles in κ cells away from the cell containing particle i in the calculation, we set $r_{e,new} = \kappa_{new} h = \sqrt{\left((\kappa+1)^2 h^2 + (\kappa+1)^2 h^2\right)} = \sqrt{2}(\kappa+1)h$. Fig. 5 shows the new influence domain (ID2). Here, only those particles lying in cells $(m-\kappa, n-\kappa) \sim (m+\kappa, n+\kappa)$ are considered in the numerical operation. This strategy is used while applying the moving least squares interpolation method to interpolate the values from moving particles to fixed grids.

In most of the particle methods, it is challenging to deal with particle clumping problem (which would cause numerical instability). Due to the fact that these particles are carrying mass information, they cannot be simply added/deleted and artificial numerical treatments (e.g. repulsive forces) are normally employed on fluid particles if clumping happens. In MPPM and our current MLE methods, particles are considered as observation points. They are simply deleted if penetration through problem domain (e.g. wall) happens or two particles clumped together (e.g. $r_{ij} \leq 0.1(\Delta x^2 + \Delta y^2)$) and added if a background mesh contains no fluid particle. In our simulations, the maximum number of fluid particles inside each cell is capped at $npcmax$, which is a prescribed simulation parameter.

4. Interpolation of solutions between Lagrangian and Eulerian mesh points

After adopting the MLE method to calculate the velocity Laplacian term, the intermediate velocity components and the locations of moving particles can be calculated by using Eqs. (3) and (4). Due to the fact that the intermediate velocity field does not satisfy the divergence-free constraint, a modified step is performed as shown in Eqs. (5) and (6). The resulting PPE is shown in Eq. (7).

From Eq. (7), the source term is the divergence of the intermediate velocity on the Cartesian grid points. Therefore, we need to interpolate the intermediate velocity components from the moving particles to the fixed grid points that store the pressure values. Here, a third-order accurate Moving Least Squares (MLS) interpolation method is adopted. A local polynomial function is constructed as

$$f(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + O(x^3, y^3) \tag{19}$$

where the coefficients a_0–a_5 are to be determined. After minimizing the error using the least squares method, the coefficients in Eq. (19) can be obtained as

$$\begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{Bmatrix} = \begin{bmatrix} \sum \omega_i^2 & \sum \omega_i^2 x_i & \sum \omega_i^2 y_i & \sum \omega_i^2 x_i^2 & \sum \omega_i^2 x_i y_i & \sum \omega_i^2 y_i^2 \\ \sum \omega_i^2 x_i & \sum \omega_i^2 x_i^2 & \sum \omega_i^2 x_i y_i & \sum \omega_i^2 x_i^3 & \sum \omega_i^2 x_i^2 y_i & \sum \omega_i^2 x_i y_i^2 \\ \sum \omega_i^2 y_i & \sum \omega_i^2 x_i y_i & \sum \omega_i^2 y_i^2 & \sum \omega_i^2 x_i^2 y_i & \sum \omega_i^2 x_i y_i^2 & \sum \omega_i^2 y_i^3 \\ \sum \omega_i^2 x_i^2 & \sum \omega_i^2 x_i^3 & \sum \omega_i^2 x_i^2 y_i & \sum \omega_i^2 x_i^4 & \sum \omega_i^2 x_i^3 y_i & \sum \omega_i^2 x_i^2 y_i^2 \\ \sum \omega_i^2 x_i y_i & \sum \omega_i^2 x_i^2 y_i & \sum \omega_i^2 x_i y_i^2 & \sum \omega_i^2 x_i^3 y_i & \sum \omega_i^2 x_i^2 y_i^2 & \sum \omega_i^2 x_i y_i^3 \\ \sum \omega_i^2 y_i^2 & \sum \omega_i^2 x_i y_i^2 & \sum \omega_i^2 y_i^3 & \sum \omega_i^2 x_i^2 y_i^2 & \sum \omega_i^2 x_i y_i^3 & \sum \omega_i^2 y_i^4 \end{bmatrix}^{-1} \times \begin{Bmatrix} \sum \omega_i^2 f_i \\ \sum \omega_i^2 f_i x_i \\ \sum \omega_i^2 f_i y_i \\ \sum \omega_i^2 f_i x_i^2 \\ \sum \omega_i^2 f_i x_i y_i \\ \sum \omega_i^2 f_i y_i^2 \end{Bmatrix} \tag{20}$$

where i is the index of the interpolated point.

Since pressure is a field variable, the PPE is solved on the fixed Eulerian Cartesian grids to retain its ellipticity. To avoid dealing with unphysical pressure boundary condition, a second-order finite volume method is adopted to solve the PPE on staggered grids. The discretized PPE is:

$$(u_e - u_w)\,\Delta y + (v_n - v_s)\,\Delta x = 0. \tag{21}$$

By replacing u and v with Eq. (5), the following equation is derived

$$\left\{\left[u_e^* - \frac{\Delta t}{\rho}\left(\frac{\partial p}{\partial x}\right)_e\right] - \left[u_w^* - \frac{\Delta t}{\rho}\left(\frac{\partial p}{\partial x}\right)_w\right]\right\}\Delta y + \left\{\left[v_n^* - \frac{\Delta t}{\rho}\left(\frac{\partial p}{\partial y}\right)_n\right] - \left[v_s^* - \frac{\Delta t}{\rho}\left(\frac{\partial p}{\partial y}\right)_s\right]\right\}\Delta x = 0. \tag{22}$$

The pressure gradient terms shown in the above equation are replaced with the second-order central difference scheme, i.e., $(\partial p/\partial x)_e = (p_E - p_P)/\Delta x$. After some algebraic manipulation, the discretized PPE can be expressed as

$$2\left(\frac{\Delta y}{\Delta x} + \frac{\Delta x}{\Delta y}\right) p_P - \frac{\Delta y}{\Delta x}(p_E + p_W) - \frac{\Delta x}{\Delta y}(p_N + p_S) = \frac{\rho}{\Delta t}\left[\left(u_w^* - u_e^*\right)\Delta y + \left(v_s^* - v_n^*\right)\Delta x\right]. \tag{23}$$

Since the resultant matrix in Eq. (23) is symmetric and positive-definite, Conjugate Gradient (CG) iterative solver is suitably applied to solve the linear system.

To compute the pressure gradient term on moving particles, a bilinear interpolation scheme is adopted to obtain

$$\left(\frac{\partial p}{\partial x_i}\right)_P = \sum_{j=1}^{4} \frac{\partial N_j}{\partial x_i} p_{G,j}. \tag{24}$$

With the bilinear shape functions introduced above, the gradient terms can be approximated as follows

$$\frac{\partial p}{\partial x} = \frac{\partial p}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{1}{\Delta x}\left(\frac{y - y_P}{\Delta y}(p_P - p_E - p_N + p_{NE}) - (p_P - p_E)\right) \tag{25}$$

$$\frac{\partial p}{\partial y} = \frac{\partial p}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial y} = \frac{1}{\Delta y}\left(\frac{x - x_P}{\Delta x}(p_P - p_E - p_N + p_{NE}) - (p_P - p_N)\right). \tag{26}$$

The neighboring pressure points used in the above two equations are shown in Fig. 6.

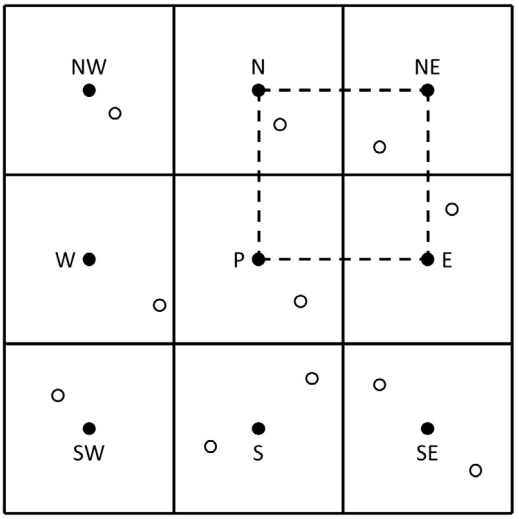


Fig. 6. Schematic of the pressure points used in the calculation of pressure gradient terms.

Within the framework of MLE method, the solution algorithm is summarized as follows:

1. Eqs. (14)–(16) are used to compute the velocity Laplacian.
2. Eqs. (3) and (4) are then solved to get the intermediate velocity components and the intermediate particle locations.
3. The intermediate velocity components are interpolated from moving particles to fixed grid points by using Eqs. (19) and (20).
4. Eq. (23) is solved to get the pressure solution.
5. Eq. (24) is used to compute pressure gradients appeared in Eqs. (5) and (6) in order to get the new velocity component and particle location. Also, the intermediate velocity components on Eulerian grid points are updated.

5. Verification studies

5.1. Couette flow

The first test case involves Couette flow motion between two infinite parallel plates located at $y = 0$ and $y = H$. While the bottom plate is a stationary one, the upper plate is driven by a velocity $(u_0, v_0) = (1.0, 0.0)$. Since an infinite flow domain is considered, periodic boundary condition is assumed in the direction of x-axis. The exact solution of this problem is:

$$u(y,t) = \frac{u_0}{H}y + \sum_{n=1}^{\infty} \frac{2u_0}{n\pi}(-1)^n \sin\left(\frac{n\pi}{H}y\right) \exp\left(-\nu\frac{n^2\pi^2}{H^2}t\right) \tag{27}$$

which was derived by Morris in [39].

The simulation is performed at $npcmax = 1$, $H = 1$, $\nu = 1$, and $u_0 = 1$ in a rectangular domain with $0 \le x \le 10$ and $0 \le y \le 1$. The computational domain and boundary conditions are schematically shown in Fig. 7. The Reynolds number is computed as:

$$Re = \frac{u_0 H}{\nu}. \tag{28}$$

In the simulation, the size of the pressure mesh is $\Delta x = \Delta y = 0.1$. The number of moving particles is 1000 which is the same as the number of pressure cells. In order to test the accuracy of the MLE scheme in approximating the

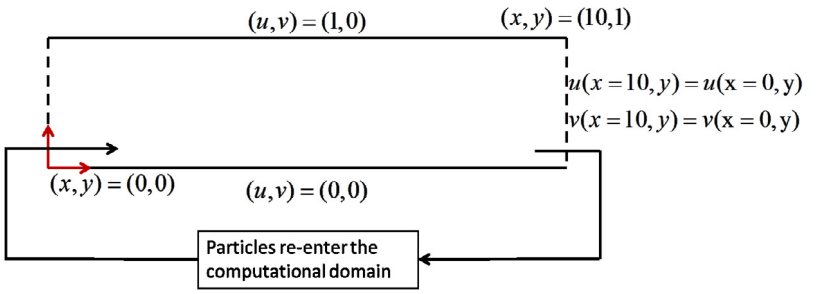
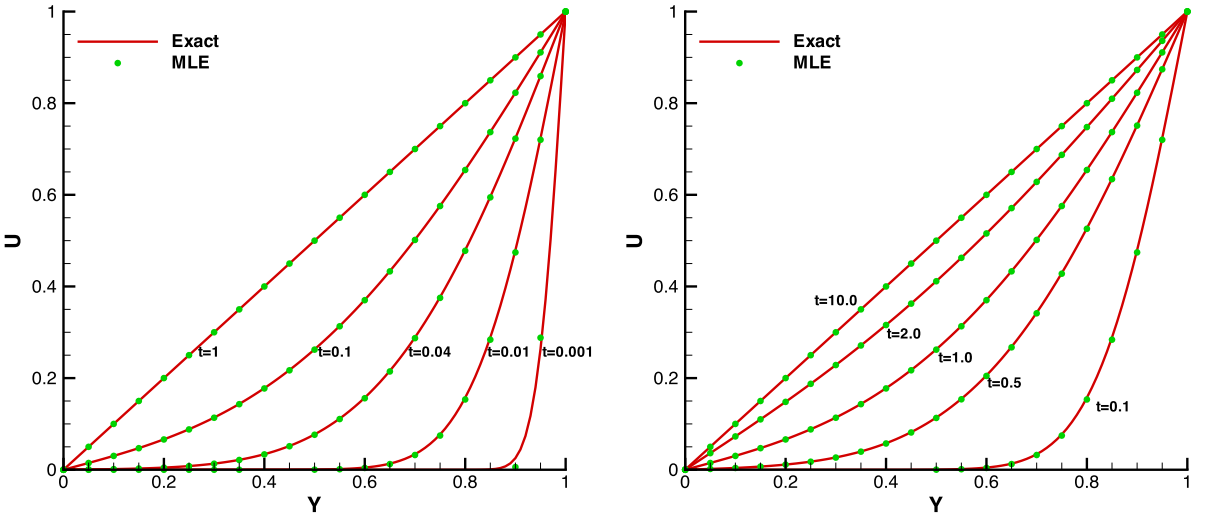


Fig. 7. Schematic of the computational domain and the boundary conditions in Couette flow.



(a) $Re = 1$. (b) $Re = 10$.

Fig. 8. The predicted time varying velocity profile $u(y, t)$ for the Couette flow (line: Exact ; dot: MLE).

velocity Laplacian, a viscous dominated Couette flow is considered here. Fig. 8(a) and 8(b) show the numerical results for $Re = 1$ and 10 at different time levels. It is appealing to note that all the computed results agree considerably well with the exact solutions.

5.2. *Plane Poiseuille flow*

The second test case deals with a plane Poiseuille flow driven by a unidirectional external force F_x. Here, the two infinite parallel plates are both stationary. The domain size is similar to that considered in the previous case. No-slip boundary conditions are applied on the upper and lower plates. Again, the left and right edges are assumed to be periodic. The schematic diagram detailing the computational domain and the boundary conditions is shown in Fig. 9. Exact solution of this problem exists in [39] as well:

$$u(y,t) = \frac{F_x}{2\nu} y\,(H-y) - \sum_{n=0}^{\infty} \frac{4F_x H^2}{\nu\pi^3(2n+1)^3} \sin\left[\frac{\pi y}{H}(2n+1)\right] \exp\left[-\frac{(2n+1)^2\pi^2\nu}{H^2}t\right]. \tag{29}$$

The maximum steady-state velocity occurs at $y = H/2$ such that

$$u_{max} = \frac{F_x}{2\nu}\frac{H}{2}\left(H - \frac{H}{2}\right) = \frac{F_x Re H^2}{8}. \tag{30}$$

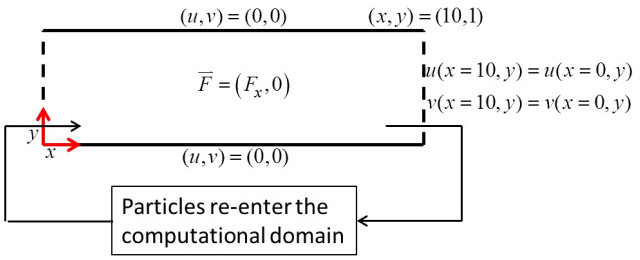
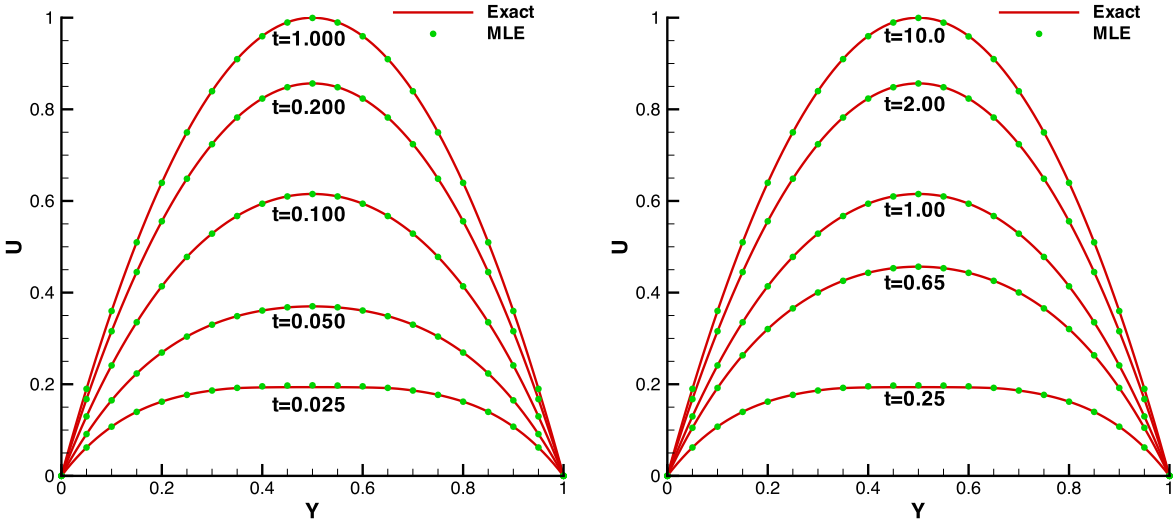


Fig. 9. Schematic of the computational domain and the boundary conditions in plane Poiseuille flow.



(a) $Re = 1$. (b) $Re = 10$.

Fig. 10. The predicted time varying velocity profile $u(y, t)$ for the plane Poiseuille flow (line: Exact ; dot: MLE).

By fixing the maximum velocity to 1.0, two viscous dominated flow cases of different Reynolds numbers, i.e. 1 ($F_x = 0.8$) and 10 ($F_x = 0.8$), are simulated. The *npcmax* is set to 1 and 2 for $Re = 1$ and 10, respectively. It is interesting to note that the predicted steady state solutions agree very well with the exact solutions as shown in Fig. 10(a) and 10(b).

5.3. Taylor–Green flow

In this section, we intend to study the spatial rate of convergence of the current MLE scheme. Here, the Taylor–Green flow problem is simulated in a unit square domain. The exact solutions of velocity components and pressure are:

$$u(x, y, t) = -U e^{bt} \cos(\pi x) \sin(\pi y) \tag{31}$$

$$v(x, y, t) = U e^{bt} \sin(\pi x) \cos(\pi y) \tag{32}$$

$$p(x, y, t) = -\frac{U^2}{4} e^{2bt} (\cos(2\pi x) + \sin(2\pi y)) \tag{33}$$

where $U = 1.0$, and $b = \frac{-2\pi^2}{Re}$. The calculations are carried out at four different grid numbers: 21^2, 41^2, 81^2, and 161^2 for the case of $Re = 100$. Simulations are executed until $t = 5$ s with the time step $\Delta t = 0.01\Delta x$ ($CFL = 0.01$). The *npcmax* is set to 1.

Table 4
Spatial rates of convergence for the case of Taylor-Green Taylor–Green flow.

Mesh size	u		v		p	
	L_2-norm	R.O.C.	L_2-norm	R.O.C.	L_2-norm	R.O.C.
5.00E−2	1.2174E−03	–	1.2477E−03	–	5.6792E−01	–
2.50E−2	3.2667E−04	1.9305	3.3536E−04	1.9289	2.7702E−01	1.4318
1.25E−2	8.4532E−05	1.9658	8.6888E−05	1.9647	1.3756E−01	1.4191
6.25E−3	2.1500E−05	1.9829	2.2115E−05	1.9821	6.8661E−02	1.4154

Table 5
CPU time for different subroutines used in simulations of Taylor-Green Taylor–Green flow with 41^2 grids. main for the whole program. Lap for calculating Laplacian terms, Pre for solving PPE, and Interp for interpolating intermediate velocities from moving particles to fixed grids.

Routine	MP		SD		GFDM		MLE	
	sec.	Proportion	sec.	Proportion	sec.	Proportion	sec.	Proportion
main	9.53	100.00%	20.20	100.00%	11.64	100.00%	8.55	100.00%
Lap	1.77	18.66%	12.13	60.07%	3.57	30.70%	0.34	4.01%
Pre	2.01	21.11%	2.37	11.74%	2.34	20.11%	2.31	27.01%
Interp	4.66	48.94%	4.46	22.08%	4.77	41.02%	4.77	55.84%

The numerical results obtained on the finest pressure mesh are compared with the exact solutions as shown in Fig. 11. To check the spatial Rate of Convergence (R.O.C.), the L_2-error norms (Eq. (34)) after the first time step are calculated in order to eliminate the possible influence of the time step size in affecting the calculation of R.O.C. [40]. Table 4 shows that the predicted velocities exhibit second order convergence. The R.O.C. of pressure, on the other hand, is about 1.4. Research is currently underway to approximate the pressure Laplacian in PPE in a more accurate manner.

$$\|\phi\|_2 = \sqrt{\frac{\sum_{i=1}^{N}(\phi_i(t) - \phi_{exact}(t))^2}{N}}. \tag{34}$$

Also, in order to demonstrate the ability of the current particle addition/deletion procedure used to prevent generally non-uniform particles distribution, Fig. 12 shows the particle distribution and the pressure field computed on 21^2 grids. As seen, only one particle exists in each cell ($npcmax = 1$) and the pressure field is smooth.

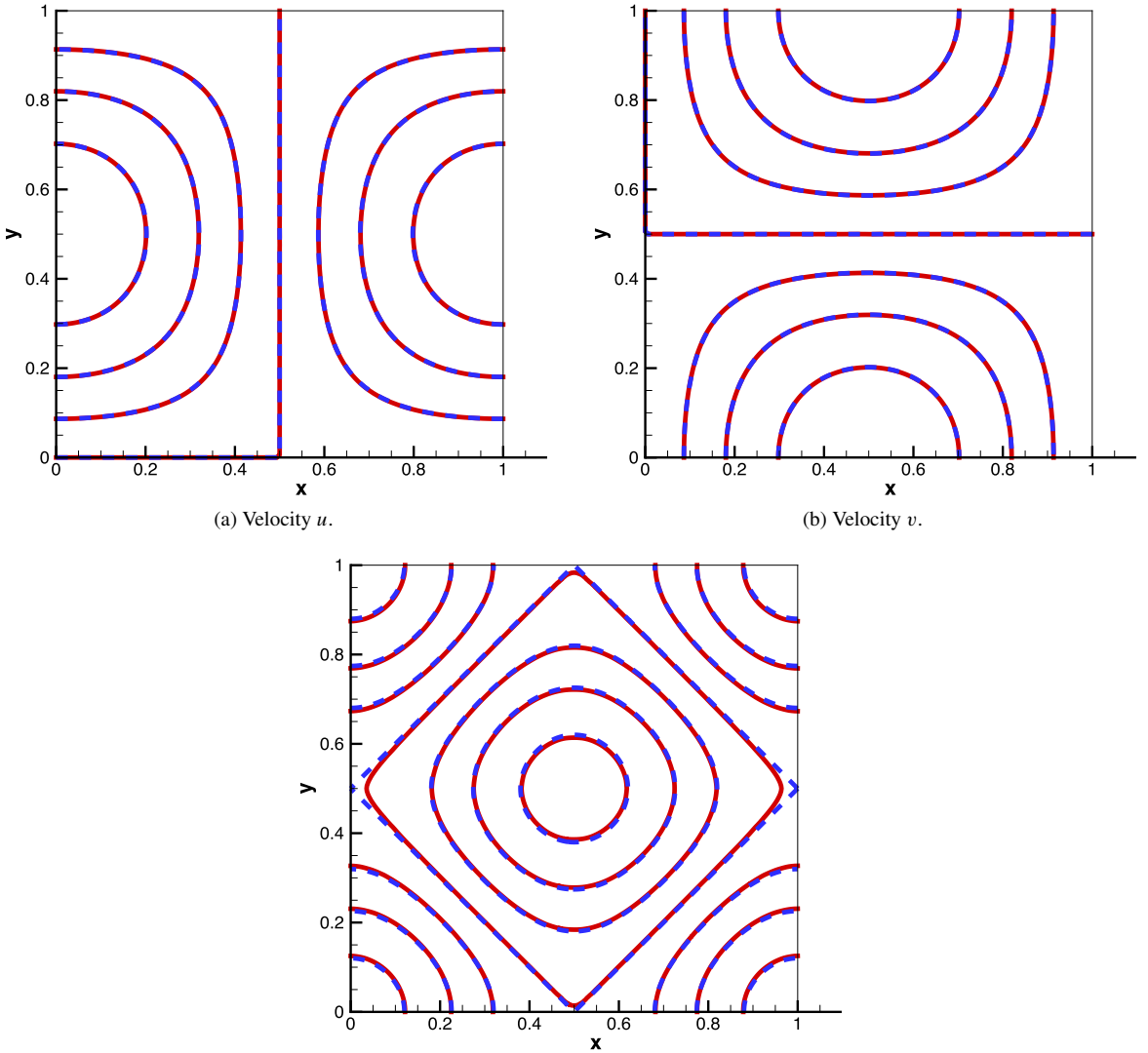
We have further validated our results obtained from [41,42,31]. Fig. 13 compares the decay of maximum velocity with the exact solution Ue^{bt}. The relative error of the maximum velocity is shown in Fig. 14. Here, the relative error is calculated via:

$$L_\infty(t) = \frac{\mid max\|\vec{u}\| - e^{bt} \mid}{e^{bt}} \times 100\%. \tag{35}$$

As compared to [42], our predicted maximum velocities are smoother and exhibit a smaller error. Meanwhile, as shown in Fig. 15, the values of $L_{\infty,max}$, i.e. the maximum value of L_∞, are compared with those predicted in [41,31] for different numbers of particles. As seen, the error of MLE undergoes a smoother decay as compared to [41] and smaller errors are resulted as compared to [31].

Next, in order to check the numerical dissipation of our new MLE method, the Taylor–Green flow problem is simulated on 201^2 grids at higher Reynolds number, i.e. $Re = 10^6$. Fig. 16 shows that our numerical results agree considerably well with the exact solutions at $t = 5$ s. The numerical dissipation is not apparently found in our MLE method because the current MLE method does not involve convective discretization.

Furthermore, in order to show the efficiency of our proposed MLE method in calculating the Laplacian terms on moving particles, the computational times for different Laplacian models and some other most time-consuming subroutines in the code are reported in Table 5. Here we choose $Re = 100$ and the simulation is performed on 41^2 grids for 5 s. It is observed that computational times for solving PPE and interpolating intermediate velocities from moving particles to fixed grids are almost the same. It is important to note that MP scheme is an explicit scheme while SD and GFDM schemes involve solving a 5×5 matrix for each particle. Therefore, the CPU time for MP scheme



(a) Velocity u.

(b) Velocity v.

(c) Pressure p.

Fig. 11. Comparison of the predicted and exact solutions for the Taylor-Green Taylor–Green flow with the grid number 161^2 at $t = 5$ s (dash: exact, solid: numerical).

is significantly shorter than those of SD and GFDM. For our proposed MLE method, in spite of the fact that CCD scheme involves solving a twin-tridiagonal system, it can be efficiently solved by the twin-tridiagonal elimination method [38]. Also, since the cell index of each particle is known, the four grid points used to interpolate Laplacian terms from fixed grids to moving particles can be easily identified without relying on the searching procedure used in the conventional particle methods.

6. Validation studies

In this section, some benchmark validation problems such as the developing flow, flow over a backward-facing step, and lid-driven cavity flow are simulated.

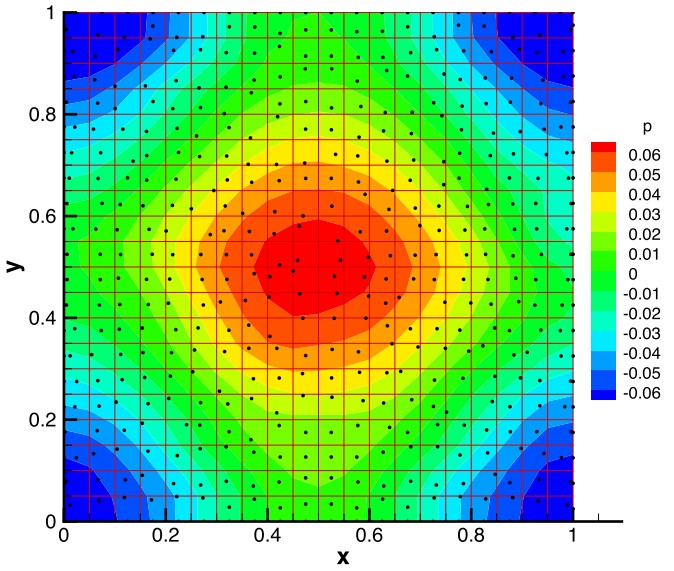
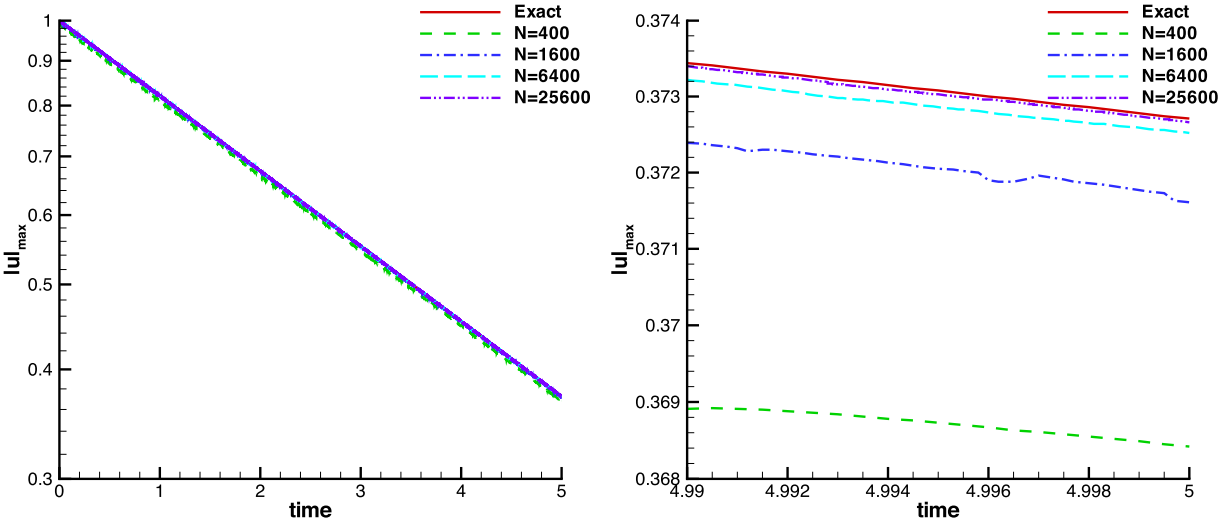


Fig. 12. Particle distribution and pressure field for Taylor-Green Taylor–Green flow computed on 21^2 grids at $Re = 100$.



(a) Whole view.

(b) Zoomed.

Fig. 13. Comparison of the exponentially decayed maximum velocities with exact solution in $t = 0$–5 s with different numbers of particles.

6.1. Developing flow problem

In this section, we intend to test the ability of MLE method to solve inflow–outflow problem. The computational domain is $0 \leq x \leq 30$ and $0 \leq y \leq 1$. At the channel inlet, a uniform velocity is prescribed as $u(x = 0, y) = 1.0$ and $v(x = 0, y) = 0.0$. No-slip boundary conditions ($u = v = 0.0$) prevail at both upper and lower boundaries. At the outlet, Neumann velocity boundary condition is adopted: $\partial u/\partial x = \partial v/\partial x = 0.0$ and the Dirichlet pressure boundary condition is enforced ($p = 0$). The schematic diagram of computational domain and the corresponding boundary

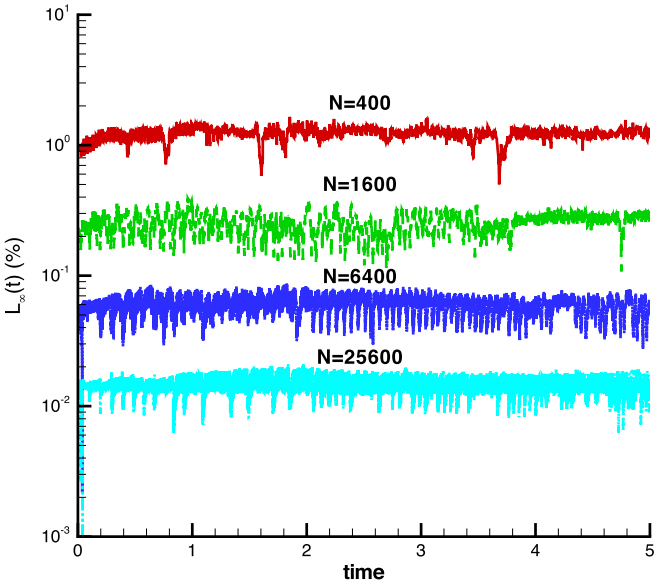


Fig. 14. The relative errors of the maximum velocity ($L_\infty(t)(\%)$) for the cases with different numbers of particles.

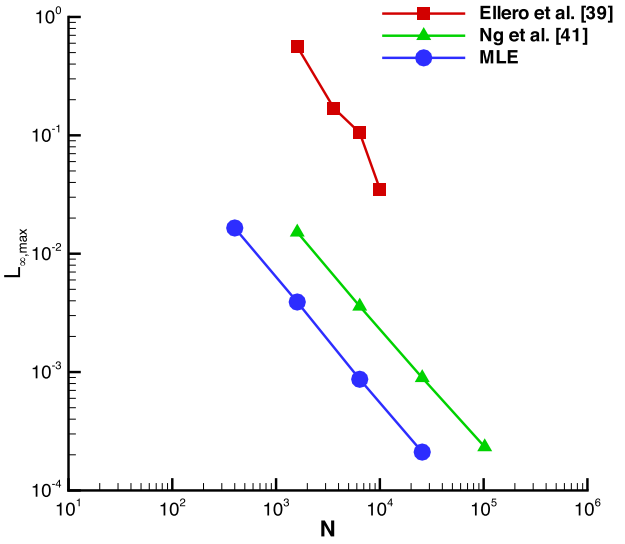


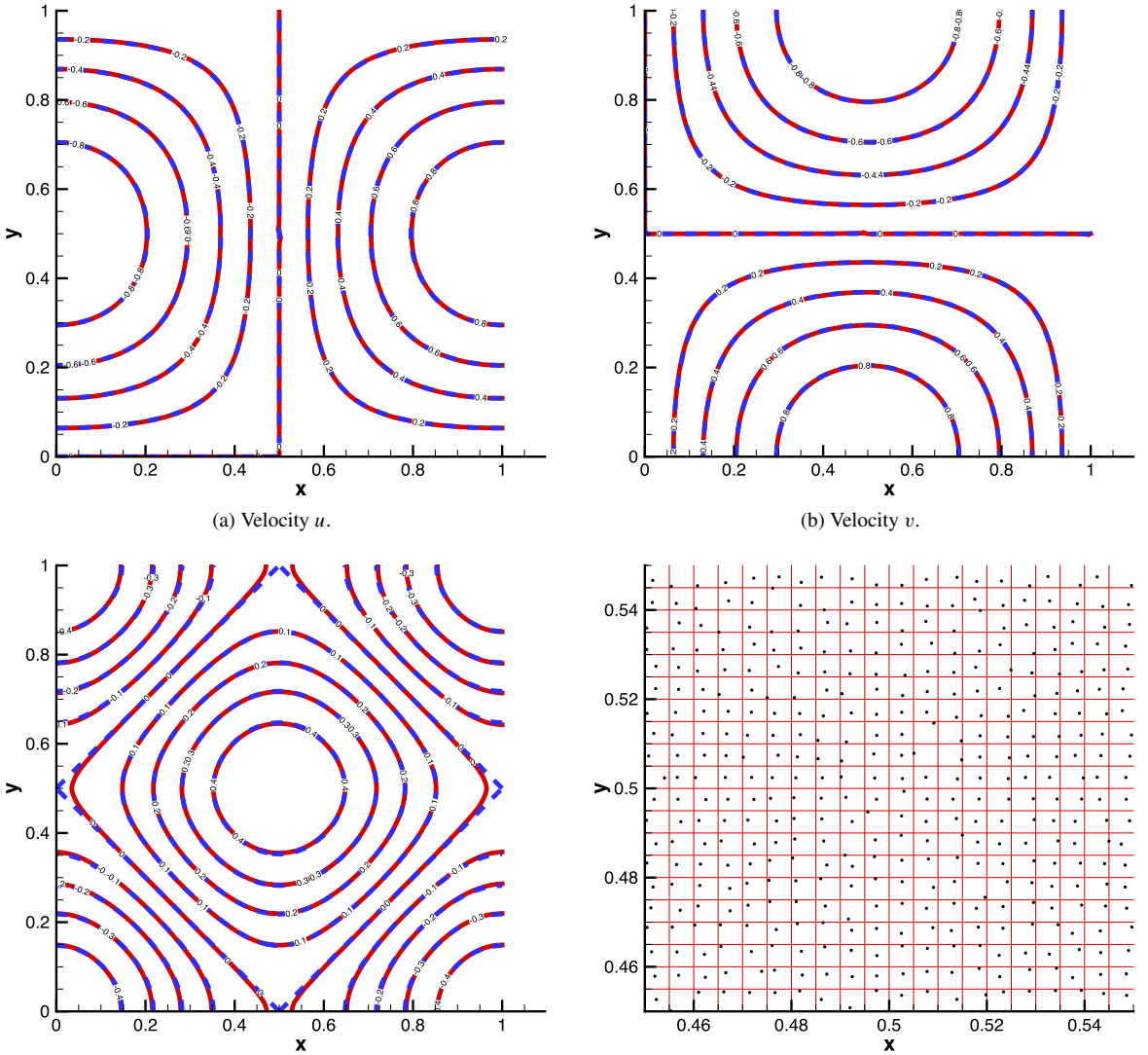
Fig. 15. Comparison of the values of $L_{\infty,max}$ under different numbers of particles.

conditions are shown in Fig. 17. The Reynolds number of this problem is defined as:

$$Re = \frac{\rho U H}{\mu} \tag{36}$$

where ρ is the density (=1), U is the inlet velocity (=1), H is the channel height, and μ is the dynamic viscosity. Here, the Reynolds number is set to 150 to mimic laminar flow condition. The $npcmax$ is set to 4. Therefore, the developing length is less than the total length of the computational domain. The grid size is $\Delta x = \Delta y = 0.05$.

To validate our numerical results, the centerline u-velocity along the flow direction is compared with the reference data obtained from [43,44,29]. Brandt et al. [43] have utilized the streamfunction–vorticity approach and Van Dyke [44] has adopted the perturbation method to obtain the asymptotic solutions. The numerical results of



(a) Velocity u.

(b) Velocity v.

(c) Pressure p.

(d) Particles distribution.

Fig. 16. Comparison of the predicted and exact solutions for the Taylor-Green Taylor–Green flow with $Re = 10^6$ and the grid number 201^2 at $t = 5$ s (dash: exact, solid: numerical).

Hwang [29], whereby the Laplacian term was discretized by using the first-order Smoothing Difference Laplacian scheme (MPPM-SD), have been also used in our comparison study. Our numerical results agree considerably well with those of [43,29] at the initial stage of flow development. However, as the flow is developing further, the MLE solutions are coming closer to those of [43]. The results in [44], on the other hand, are not comparable to our predictions and the reference solutions in [43].

The cross-sectional u-velocity profile at $x = 30$ is compared with the exact solution of the fully-developed flow as well:

$$u_{fully-developed}(y) = 6U_{in}y(1 - y). \tag{37}$$

The maximum u at $y = 0.5$ is 1.50926911 m/s $\sim$ 1.5 m/s (theoretical solution). Fig. 18 shows that the MLE solutions agree considerable well with the exact solutions as the relative error is about 0.62%.

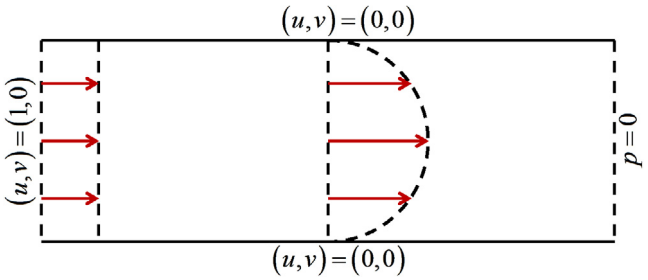
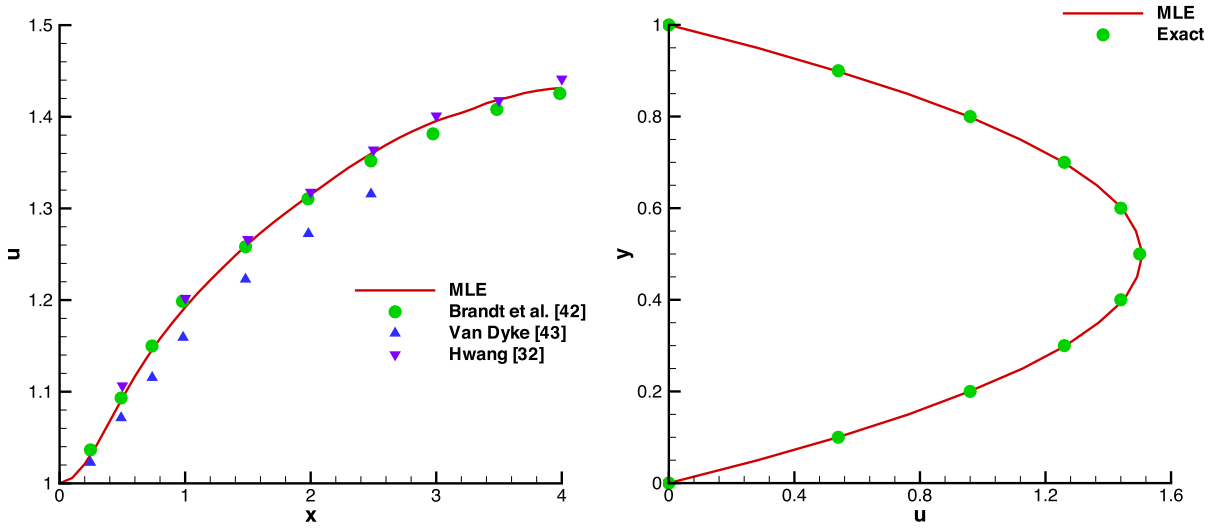


Fig. 17. Schematic of the developing flow problem.



(a) Comparison of the predicted centerline u-velocities at inlet region.

(b) Comparison of the predicted and exact velocity profiles for u at $x = 30$.

Fig. 18. Comparisons of the predicted velocity profiles with other numerical and exact solutions.

6.2. Backward-facing step problem

This problem involves flow separation and reattachment. Armaly et al. [45] previously investigated this problem theoretically and experimentally. They have designed a channel upstream of the step region in order to achieve a uniform flow condition. Here, a fully-developed inflow velocity condition is considered for simplicity. The computational domain and boundary conditions are schematically shown in Fig. 19, where the step height is $H_s = 4.9$, the inlet channel height is $H_i = 5.2$, and the average inflow velocity U is 1.0. The pressure mesh size of $\Delta x = \Delta y = 0.1$ is employed. The Reynolds number ($Re = 2UH_i/\nu$) is 100, where the density ρ is taken as 1.0. The total simulation time is 100.0 s. The *npcmax* is set to 1.

To compare with the reference data given in [45,29,31], the u-velocity component (normalized by U) at different locations (normalized by H_s) are plotted in Fig. 20. The velocity profile at the location $x/H_s = 0.00$ is seen to deviate slightly from the referenced data because the upstream inlet channel has been omitted in the current study. At $x/H_s = 2.55$, the predicted axial velocities obtained from the MLE method and the original MPPM [29] are comparable. Both methods are able to capture the recirculation flow near the step region. Meanwhile, as the fluid becomes fully-developed ($x/H_s = 5.41$), it is interesting to note that the MLE results tend to be closer to those measured. Seemingly, the axial velocity gradient near the bottom wall has been over-predicted in [29]. Downstream of the channel inlet, our MLE predictions match considerably well with the recent UMPPM solutions [31].

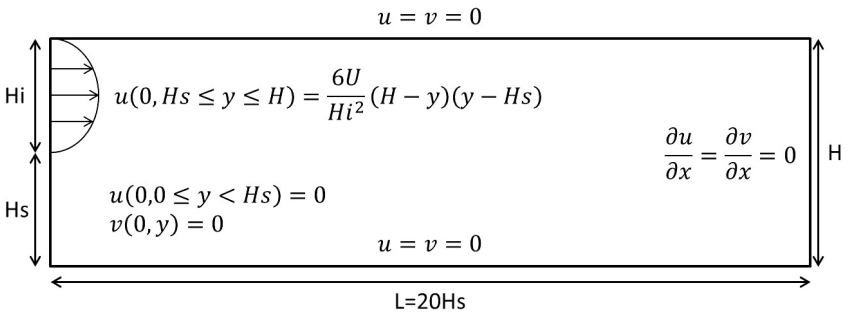


Fig. 19. Schematic of the backward-facing step flow problem.

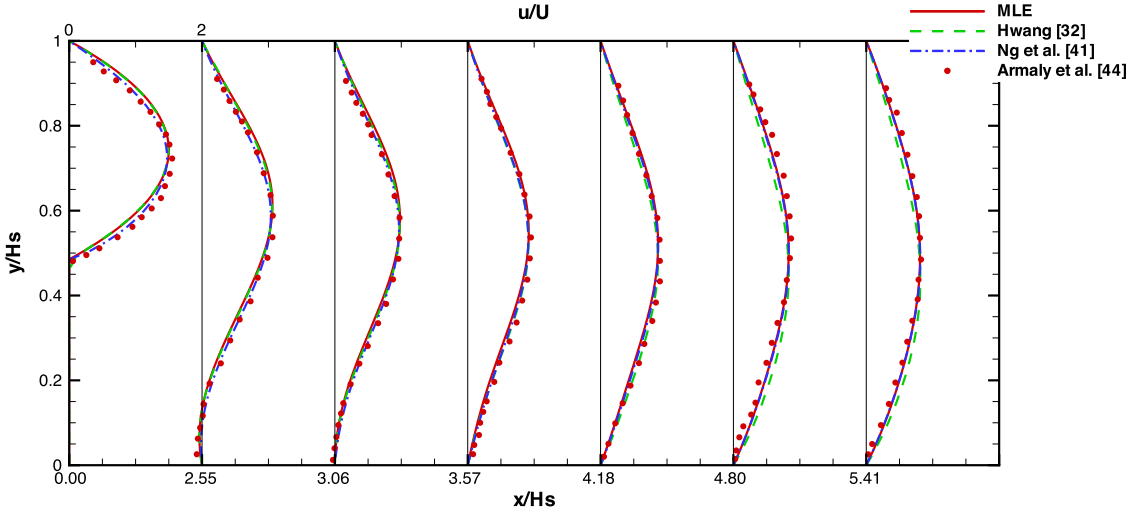


Fig. 20. The predicted velocity profiles for the backward-facing step flow.

6.3. Lid-driven cavity flow problem

The computational domain is a unit square and the top boundary is driven by a velocity vector $(u_0, v_0) = (1, 0)$. The main flow structure is a primary vortex generated near the center of the domain. As the Reynolds number increases, corner vortices emerge and become larger in size. Meanwhile, the flow gradient becomes more pronounced, posing numerical challenges in most of the numerical methods. Thus, such a classical benchmark problem is always used to validate newly-developed numerical methods.

The computational domain, boundary conditions, and vortices are schematically shown in Fig. 21. Here, density ρ, characteristic velocity U and length L are set to unity. The dynamic viscosity is computed as $\mu = \rho U L/Re$. To test the ability of particle method in overcoming the problem of convective instability, the cases with Reynolds numbers ranging from 100 to 10 000 are conducted.

In order to compare our results with those of MPPM-SD [29], the cases of $Re = 100$, 400, and 1000 are simulated by using 41^2 grid points ($\Delta x = \Delta y = 1/40$). For cases of higher Reynolds numbers, i.e. $Re = 5000$, 7500, and 10 000, a total number of 81^2 grid points ($\Delta x = \Delta y = 1/80$) is used. In order to obtain a statistically steady results, the simulation times for the cases of $Re = 100$, 400, and 1000 are 100 s, 100 s, and 200 s, respectively. The simulation time is set to 1000 s for cases with higher values of Re. The *npcmax* is set to 4.

Next, the velocities at the centerlines of the square cavity are compared with the reference solutions of Ghia et al. [46] which were computed by using the streamfunction–vorticity formulations on dense grids. As shown in Fig. 22, the current MLE and MPPM-SD [29] results are hardly discernible at $Re = 100$, and 400. However, when Re

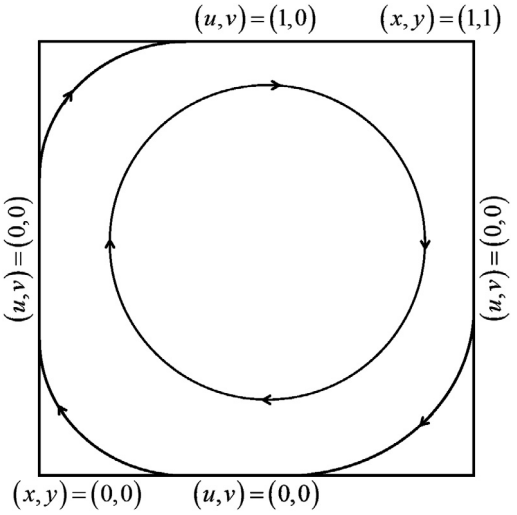
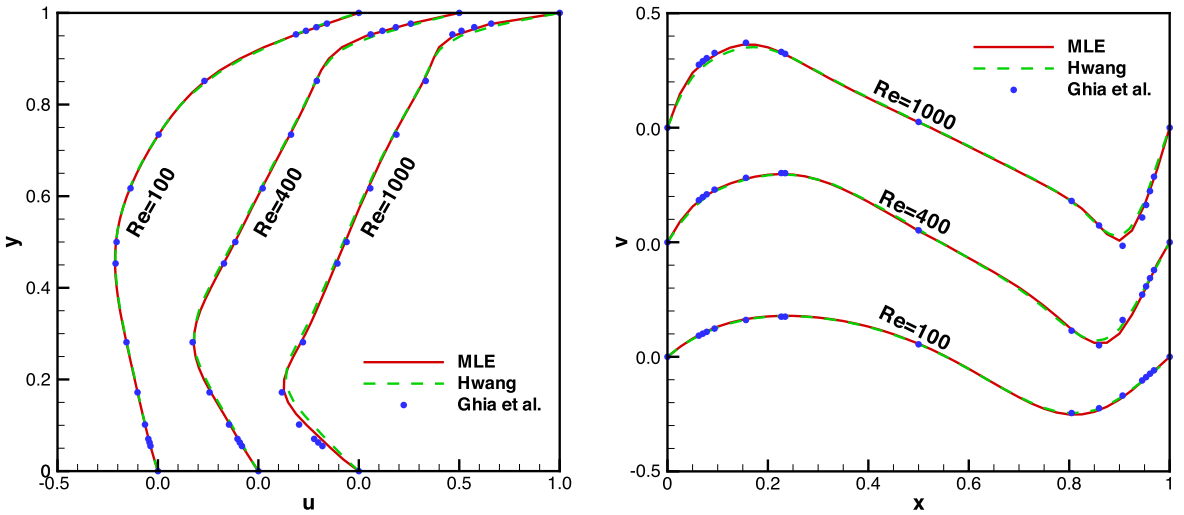


Fig. 21. Schematic of the boundary conditions and vortices formed in the lid-driven cavity.

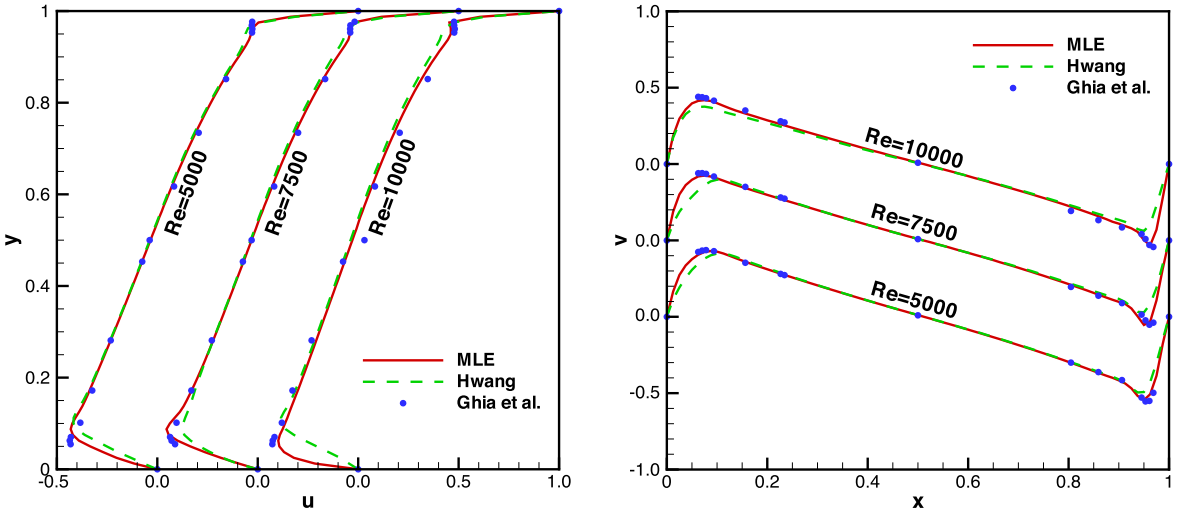


(a) Velocity u. (b) Velocity v.

Fig. 22. The predicted velocity profiles for the lid-driven cavity flow at low Reynolds numbers with 41^2 grids (solid: MLE, dashed: Hwang [29], dot: Ghia et al. [46]).

is increased to 1000, the u-velocity gradient at the bottom wall can be resolved reasonably well by the current MLE method. The results obtained from MPPM-SD method, on the other hand, has exhibited under-predicted u-velocity gradients near the walls. As shown in Fig. 23, the degree of under-prediction is more evident for MPPM-SD as Re increases. Meanwhile, it is appealing to note that the current MLE method is able to capture the sharp velocity gradient well even on the current mesh resolution which is considerably coarser than that of Ghia et al. [46].

Moreover, we have compared our results with those of SPH reported by Lee et al. [47] for $Re = 400$ and 1000 (c.f. Figs. 5 and 6 in [47]). In general, the solutions of Weakly-Compressible SPH (WCSPH) are wiggling. This problem can be resolved by using Incompressible SPH (ISPH) method as demonstrated by [47]. However, by comparing the MLE and ISPH solutions computed on the same mesh resolution of $\Delta x = \Delta y = 1/40$, it is appealing to point out that the current MLE method is able to produce a more accurate velocity profile than the ISPH method.



(a) Velocity u. (b) Velocity v.

Fig. 23. The predicted velocity profiles for the lid-driven cavity flow at high Reynolds numbers with 81^2 grids (solid: MLE, dashed: Hwang [29], dot: Ghia et al. [46]).

7. Concluding remarks

In this paper, we have proposed an improved MPPM method known as the Mixed Lagrangian–Eulerian (MLE) method. The spatial derivative terms are calculated on the Cartesian mesh to attain higher order accuracy and the results are interpolated to Lagrangian particles. Meanwhile, the advection term is treated by using the moving particle method to get rid of the convective instability problem. No ghost particles are employed in the current work, thereby avoiding errors generated from some unphysical implementations. Also, we have proposed a proper means to calculate the radius of influence domain and particle chosen strategy to make sure particles have been distributed uniformly, thanks to the particle addition and deletion. The moving least squares interpolation method has been adopted to interpolate values from moving particles to fixed grids.

Our numerical results obtained from various benchmark problems compare very well with the reference solutions. Although the calculation time for Laplacian operator is significantly reduced, the particle–mesh interpolation procedure requires a relatively large computational effort. Nevertheless, since our background pressure mesh is fixed, a significant saving in computational time can be realized because the coefficients of PPE remain unchanged in the current MLE method (unlike the ISPH and MPS methods). The only tuning parameter introduced in our method is the maximum number of particles in each cell (i.e. $npcmax$). From the simulations, we have found that flow accuracy can be improved by simply increasing the number of $npcmax$ without increasing the number of pressure mesh.

Acknowledgment

This work was supported by the Ministry of Science and Technology (MOST) of Republic of China (R.O.C.) under the Grants MOST105-2221-E-002-066.

References

[1] B.J. Daly, F.H. Harlow, J.E. Welch, E.E. Sanmann, Numerical fluid dynamics using the particle and force method, Los Alamos Scientific Laboratory (1965) LA–3144.
[2] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical starts, Mon. Not. R. Astron. Soc. 181 (1977) 375–389.
[3] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, Astronom. J. 82 (1977) 1013–1024.
[4] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, Comput. Methods Appl. Mech. Engrg. 139 (1996) 3–47.
[5] G.R. Liu, M.B. Liu, Smoothed Particle Hydrodynamics: A Meshfree Particle Method, World Scientific Publishing Co. Pte. Ltd., 2003.

[6] S. Li, W.K. Liu, Meshfree and particle methods and their applications, Appl. Mech. Rev. 55 (2002) 1–34.
[7] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nucl. Sci. Eng. 123 (1996) 421–434.
[8] M. Kondo, S. Koshizuka, Improvment of stability in improving particle semi-implicit method, Internat. J. Numer. Methods Fluids 65 (2011) 638–654.
[9] A. Khayyer, H. Gotoh, Modified moving particle semi-implicit methods for the prediction of 2D wave impact pressure, Coast. Eng. 56 (2009) 419–440.
[10] M. Tanaka, T. Masunaga, Stabilization and smoothing of pressure in MPS method by quasi-compressibility, J. Comput. Phys. 229 (2010) 4279–4290.
[11] A. Khayyer, H. Gotoh, Enhancement of stability and accuracy of the moving particle semi-implicit method, J. Comput. Phys. 230 (2011) 3093–3118.
[12] K. Shibata, I. Masaie, M. Kondo, K. Murotani, S. Koshizuka, Improved pressure calculation for the moving particle semi-implicit method, Comput. Part. Mech. 2 (2015) 91–108.
[13] J. Sanchez-Mondragon, On the stabilization of unphysical pressure oscillations in MPS method simulations, Internat. J. Numer. Methods Fluids 82 (2016) 471–492.
[14] J.J. Monaghan, Simulating free surface flows with SPH, J. Comput. Phys. 110 (1994) 399–406.
[15] S. Cummins, M. Rudman, An SPH projection method, J. Comput. Phys. 152 (1999) 584–607.
[16] S. Shao, E.Y.M. Lo, Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface, Adv. Water Resour. 26 (2003) 787–800.
[17] F. Colin, R. Egli, F.Y. Lin, Computing a null divergence velocity field using smoothed particle hydrodynamics, J. Comput. Phys. 217 (2006) 680–692.
[18] S. Shao, H. Gotoh, Turbulence particle models for tracking free surfaces, J. Hydraul. Res. 43 (2010) 276–289.
[19] J.T. Batina, A gridless Euler/Navier–Stokes solution algorithm for complex-aircraft applications, NASA Langley Research Center (1992) AIAA 93–0333.
[20] N. Tanaka, The CIVA method for mesh-free approaches: Improvement of the CIP method for n-simplex, Comput. Fluid Dyn. J. 8 (1999) 121–127.
[21] H.Y. Yoon, S. Koshizuka, Y. Oka, A particle-gridless hybrid method for incompressible flows, Internat. J. Numer. Methods Fluids 30 (1999) 407–424.
[22] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: Diffuse approximation and diffuse elements, Comput. Mech. 10 (1992) 307–318.
[23] E. Onate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, A finite point method in computational mechanics. Applications to convective transport and fluid flow, Internat. J. Numer. Methods Engrg. 39 (1996) 3839–3866.
[24] S.N. Atluri, T. Zhu, A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics, Comput. Mech. 22 (1998) 117–127.
[25] E.J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - I surface approximations and partial dirivative estimates, Comput. Math. Appl. 19 (1990) 127–145.
[26] E.J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - II solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput. Math. Appl. 19 (1990) 147–161.
[27] T. Zhu, J. Zhang, S.N. Alturi, A meshless numerical method based on the local boundary integral equation (LBIE) to solve linear and non-linear boundary value problems, Eng. Anal. Bound. Elem. 23 (1999) 375–389.
[28] Y.H. Hwang, A moving particle method with embedded pressure mesh (MPPM) for incompressible flow calculations, Numer. Heat Transfer B 60 (2011) 370–398.
[29] Y.H. Hwang, Smoothing difference scheme in a moving particle method, Numer. Heat Transfer B 60 (2011) 203–234.
[30] K.C. Ng, Y.H. Hwang, T.W.H. Sheu, C.H. Yu, Moving particle level-set (MPLS) method for incompressible multiphase flow computation, Comput. Phys. Comm. 196 (2015) 317–334.
[31] K.C. Ng, T.W.H. Sheu, Y.H. Hwang, Unstructured moving particle pressure mesh (UMPPM) method for incompressible isothermal and non-isothermal flow computation, Comput. Methods Appl. Mech. Engrg. 305 (2016) 703–738.
[32] N. Moes, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, Internat. J. Numer. Methods Engrg. 46 (1999) 131–150.
[33] S. Zhang, K. Morita, K. Fukuda, N. Shirakawa, An improved MPS method for numerical simulations of convective heat transfer problems, Internat. J. Numer. Methods Fluids 51 (2006) 31–47.
[34] C.L. Huang, T.W.H. Sheu, T. Ishikawa, T. Yamaguchi, Development of a particle interaction kernel for convection–diffusion scalar transport equation, Numer. Heat Transfer B 60 (2011) 96–115.
[35] Y.H. Hwang, Assessment of diffusion operators in a noval moving particle method, Numer. Heat Transfer B 61 (2012) 329–368.
[36] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Comput. Struct. 11 (1980) 83–95.
[37] C.G. Koh, M. Gao, C. Luo, A new particle method for simulation of incompressible free surface flow problems, Internat. J. Numer. Methods Engrg. 89 (2012) 1582–1604.
[38] P.C. Chu, C.W. Fan, A three-point combined compact difference scheme, J. Comput. Phys. 140 (1998) 370–399.
[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] N.J. Quinlan, L. Lobovsky, R.M. Nestor, Development of the meshless finite volume particle method with exact and efficient calculation of interparticle area, Comput. Phys. Comm. 185 (2014) 1554–1563.
[41] M. Ellero, M. Serrano, P. Espanol, Incompressible smoothed particle hydrodynamics, J. Comput. Phys. 226 (2007) 1731–1752.

[42] S. Adami, X.Y. Hu, N.A. Adams, A transport-velocity formulation for smoothed particle hydrodynamics, J. Comput. Phys. 241 (2013) 292–307.
[43] A. Brandt, J. Gillis, Magnetohydrodynamic flow in the inlet region of a straight channel, Phys. Fluids 9 (1966) 690–699.
[44] M.V. Dyke, Entry flow in a channel, J. Fluid Mech. 44 (1970) 813–823.
[45] B.F. Armaly, F. Durst, J.C.F. Pereira, B. Schonung, Experimental and theoretical investigation of backward-facing step flow, J. Fluid Mech. 127 (1983) 473–496.
[46] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[47] E.S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, P. Stansby, Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free method, J. Comput. Phys. 227 (2008) 8417–8436.