



A new high-order particle method for solving high Reynolds number incompressible flows

Rex Kuan-Shuo Liu¹ · Khai-Ching Ng² · Tony Wen-Hann Sheu^{1,3,4}

Received: 19 July 2018 / Revised: 27 September 2018 / Accepted: 30 November 2018
© OWZ 2018

Abstract

In this study, a new high-order particle method is proposed to solve the incompressible Navier–Stokes equations. The proposed method combines the advantages of particle and mesh methods to approximate the total and the spatial derivative terms under the Lagrangian and the Eulerian frameworks. Our aim is to avoid convective instability and increase solution accuracy at the same time. Data transfer from Lagrangian particles to Eulerian grids is realized by moving least squares interpolation. In contrast to the previously proposed method, there is no need to interpolate diffusion terms from Eulerian grids to Lagrangian particles. Therefore, the accuracy of the present solution will not be deteriorated by interpolation error. Additionally, no extra work is required to manage particles for searching procedure. Because no convection term needs to be discretized by upwinding schemes, false diffusion and dispersion errors will not be introduced, thereby increasing the solution accuracy. To verify the proposed particle method, several benchmark problems are solved to show that the present simulation is more stable, accurate, and efficient. The proposed particle method renders fourth- and second-order accurate solutions in space for velocity and pressure, respectively.

Keywords High-order particle method · Incompressible Navier–Stokes equations · False diffusion error · Numerical dispersion error

1 Introduction

Conventionally, there are two major classes of numerical methods for solving the incompressible Navier–Stokes equations: mesh and meshless (particle) methods. Finite difference method (FDM), finite volume method (FVM) and finite element method (FEM) are categorized as the mesh methods. They are developed based on the Taylor series expansion, conservation in a control volume, and variational principles, respectively. On the other hand, the moving particle

semi-implicit (MPS) and smoothed particle hydrodynamics (SPH) methods are the typical particle methods. MPS adopts the idea of particle interactions to model spatial differential operators, while Sph makes use of the concept of integral representation of a function and exploits the derivative of the kernel function to approximate spatial differential operators.

For the two major particle methods, MPS method was developed in 1996 [1] to deal with incompressible free surface flow problem and SPH method was firstly developed to solve astrophysics problems [2,3] in 1977. MPS solves the incompressible flows in a strong form and approximates differential operators within particle interaction framework. MPS adopts projection method to solve the Navier–Stokes equations and uses the concept of particle number density (PND) to model incompressibility [1]. However, the original particle methods suffer from the oscillating pressure fields. Several approaches have been proposed to tackle this difficulty [4–7]. Additionally, MPS method was also improved to tackle physically more complex flow problems such as the multi-resolution problem [8], multi-phase flow problem [9,10], and progressive water wave problem [11]. On the other hand, SPH solves the weakly compressible flow in

✉ Tony Wen-Hann Sheu
twhsheu@ntu.edu.tw

¹ Department of Engineering Science and Ocean engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan

² School of Engineering, Taylor’s University, Taylor’s Lakeside Campus, No. 1, Jalan Taylor’s, 47500 Subang Jaya, Selangor Darul Ehsan, Malaysia

³ Institute of Applied Mathematical Sciences, National Taiwan University, Taipei, Taiwan

⁴ Center for Advanced Study in Theoretical Sciences, National Taiwan University, Taipei, Taiwan

a weak form originally and is adopted to solve the incompressible flows [12–15]. SPH has also been applied to solve multi-resolution problem [16,17] or flexible fibers–viscous fluids interaction [18]. Both MPS and SPH methods suffer from the drawback of low accuracy and oscillatory pressure. Recently, accuracy has been improved by taking least squares or local polynomial reconstruction concepts into account [8,19,20], while pressure oscillations were eliminated by modifying the source terms in the pressure Poisson equation (PPE) [21–23]. Since PPE is an elliptic equation, Hwang proposed moving particle method with an embedded pressure mesh (MPPM) [4] to reduce pressure oscillations. Also, MPPM method was extended to solve multi-phase flow [10] or flows in complex flow domain [24]. In some of the recently proposed MPS and SPH methods, [20,25] and [26] show first-, second- and third-order accurate solutions for velocity, respectively. However, for the most important issue, they did not improve the accuracy order for pressure, which is most likely first-order accurate. This motivates us to develop a new particle method aiming primarily at increasing the spatial accuracy order for pressure to second. Moreover, the spatial accuracy order for velocity is improved to fourth as well.

In order to resolve low accuracy and inconsistency issues encountered in the conventional SPH and MPS methods, several refined methods have been proposed such as the updated Lagrangian particle hydrodynamics (ULPH) method [27], material point method (MPM) [28,29], corrective smoothed particle method (CSPM) [30], kernel gradient correction (KGC) method [31,32], finite particle method (FPM) [33], and decoupled finite particle method (DFPM) [34]. Most of these particle methods, except the MPM, are aimed at providing alternative ways, which are consistent and have higher accuracy compared with the conventional SPH and MPS methods, to approximate the gradient and the divergence operators. Also, it is noted that these numerical methods are actually solving weakly compressible Navier–Stokes equations such that the equation of state or a pressure transient equation shall be invoked for the calculation of pressure. In ULPH method, new formulations were proposed, while the others were constructed through the correction on the conventional SPH method. The accuracy order of velocity and pressure of these methods does not exceed second and first orders, respectively. In addition, ghost particles are adopted in these methods to model Dirichlet and Neumann boundary conditions for velocity. However, the proposed IMLE method exploits the advantage of the Eulerian method such that the spatial derivative terms can be approximated to yield an even higher accuracy order, e.g., sixth- and fifth-order accuracy for the pressure gradient and the velocity Laplacian terms, respectively. From the numerical tests, IMLE method shows fourth- and second-order accuracy for velocity and pressure fields, respectively. Also, with the aid of the interpolation procedure, there is no longer any need to take care of particle

penetrating across the wall. While the ULPH, CSPM, KGC, FPM, and DFPM are classified to be the fully Lagrangian methods, the MPM is also a hybrid method which combines the advantages of both Lagrangian and Eulerian methods such as the newly proposed IMLE method. Some key differences are that the primitive variables \mathbf{u} and p are transferred in two ways, namely they are interpolated from Lagrangian particles to Eulerian grids and in reverse at the integer time step levels n and $n + 1$. On the other hand, the IMLE method only interpolates the intermediate velocities from Lagrangian particles to Eulerian grids. It is noted that the source term of the PPE is calculated from the interpolated velocities. As a result, it is better to use the pressure gradient term to modify the interpolated velocities on Eulerian grids only but not those on Lagrangian particles. Therefore, at the beginning of the next time step, the velocity Laplacian term on Eulerian grids can be approximated using a higher accurate numerical scheme without adopting any interpolation procedure which will deteriorate the accuracy order. It is also noted that in the proposed IMLE method, unlike some other particle methods [29,31,33], no free parameters will be involved.

In the conventional finite difference and finite volume methods, the velocity terms are solved from the momentum equations which involve convection terms. When discretizing convection terms, upwinding scheme can effectively get rid of convective instability problem [35]. However, upwinding schemes introduce errors which can deteriorate the accuracy of the results. Firstly, discretizing convection terms shall introduce numerical dispersion error. Several dispersion-relation-preserving (DRP) schemes proposed in [35–38] can be applied to reduce such an error. Secondly, in most of the multi-dimensional flow situation, streamlines in a flow field are not aligned with the grid lines. Therefore, it is difficult to eliminate false diffusion error. Even though some numerical methods [39–42] can partly resolve this problem, within the framework of Eulerian approach it is still an open issue in the computational fluid dynamics (CFD) community.

The goal of this paper is aimed at developing an improved mixed Lagrangian–Eulerian (IMLE) method based on the previously proposed MLE method [43] to improve computational accuracy, enhance numerical stability and computational efficiency by combining advantages of both particle (meshless) and mesh methods. Diffusion terms present in the momentum equations are discretized on Eulerian grids to get high-order accuracy. It is emphasized that it is easier to get a better accuracy on Cartesian grid rather than on Lagrangian particles. Total derivative terms are approximated by simply advecting Lagrangian particles so that the procedure of discretizing convection terms can be avoided, thereby completely eliminating the convective instability problem. With the aid of the moving least squares (MLS) interpolation method [44], velocity components can be interpolated back to the Eulerian grid to calculate the source

term of the PPE. It is noted that MLS method was recently adopted in high-order methods to solve fluid flow problems [45–48]. While the PPE is solved on the Eulerian grid which is suitable for elliptic property of PPE, pressure field shows no oscillations, thus eliminating a major source of numerical error generated in conventional particle methods. In this newly proposed IMLE method, as the most outstanding feature of Lagrangian particle methods, there are no numerical dispersion and dissipative errors since no upwinding schemes for the discretization of convection terms have been adopted. Additionally, accuracy order in calculating diffusion terms is as high as that of the numerical schemes adopted on Eulerian grids and is not restricted by the interpolation scheme presented in the original MLE method.

The rest of this paper is organized as follows. Section 2 describes the governing equations and boundary conditions prescribed in this paper. In Section 3, the previously proposed mixed Lagrangian–Eulerian (MLE) method [43] is reviewed. Then the spatial discretization schemes and the proposed improved mixed Lagrangian–Eulerian (IMLE) method are detailed. Section 4 presents numerical results obtained from some classical benchmark problems including the Taylor–Green vortex, the backward-facing step, and the lid-driven cavity flow problems. Conclusions are drawn in Sect. 5.

2 Governing equations

In this study, we consider the equations for two-dimensional incompressible fluid flow. Governing equations, namely continuity and Navier–Stokes equations, are solved in the physical domain Ω whose boundary is $\partial\Omega$.

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

$$\frac{d\mathbf{u}}{dt} \equiv \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \tag{2}$$

In the above equations, \mathbf{u} is the velocity vector, p the pressure, ρ the density, and ν the kinematic viscosity. Reynolds number is defined as $Re = \frac{\rho UL}{\mu}$, where U is the characteristic velocity, L the characteristic length, and μ the dynamic viscosity. The solutions of Eqs. (1) and (2) are sought subjected to the initial condition for velocity specified in Ω and the boundary conditions for velocity and pressure prescribed on $\partial\Omega$.

$$\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega \tag{3}$$

$$\alpha\phi + \beta \mathbf{n} \cdot \nabla \phi = \gamma \quad \text{on } \partial\Omega \tag{4}$$

In Eq. (4), ϕ can be velocity \mathbf{u} or pressure p and \mathbf{n} denotes the unit outward normal vector of the boundary $\partial\Omega$. For Dirichlet

and Neumann boundary conditions, (α, β) equals $(1, 0)$ and $(0, 1)$, respectively.

3 Numerical methods

3.1 Review of the previously proposed mixed Lagrangian–Eulerian method

In the previously proposed Mixed Lagrangian–Eulerian (MLE) method [43], the pressure oscillation problem usually encountered in the conventional particle methods, such as MPS [1,5,7,21,22,49–52] and SPH [2,3,12,53–55], has been resolved by solving the elliptic PPE on Eulerian grids. Moreover, the accuracy order of discretizing the diffusion terms on the Lagrangian particles has been improved from the zeroth [56] and first [57] orders to the second order. The key idea is that the diffusion terms were firstly calculated on Eulerian grids by using the high-order scheme given in [58] and then they were interpolated to the Lagrangian particles by second-order accurate bilinear interpolation. Thus, the accuracy order of approximating diffusion terms at the locations of moving particles is restricted by the adopted interpolation scheme. After the step of advecting particles, intermediate velocities at the locations of moving particles are interpolated back to the Eulerian grids by using third-order MLS in order to calculate the source term shown in PPE. In the final step, the values of velocity components on the moving particles are updated by the pressure gradient term and these particles continue their journey again according to the final velocities.

In the original MLE method, the continuity equation is solved on Eulerian grids aiming to retain the elliptic property of the PPE. It is implied that the moving particles are regarded as observation points rather than mass points as usually used in the MPS and SPH methods. Because the particles are considered as mass points, they cannot penetrate physical boundaries and will generate less accurate or totally unacceptable solutions when particles become clustered. On the other hand, observation particles can be deleted from or added into a cell according to the particle distribution situation. For the case of highly clustered particles, the particle nearest the cell center will be kept and the others be deleted. On the contrary, a particle will be added into a cell without any particle. Also, for the balance of the computational efficiency and accuracy, a tuned parameter was proposed to control the maximum number of particles in a cell. In Sect. 4.3, numerical results obtained from the case of different values of tuned parameters in MLE methods will be compared with the proposed IMLE method.

The numerical procedure of the MLE method is shown below.

Algorithm 1 Algorithm of the MLE method

- 1: Read input file
- 2: Calculate computational parameters
- 3: **for** $time < MaxTime$ **do**
- 4: Calculate diffusion terms on Eulerian grids
- 5: Interpolate diffusion terms from Eulerian grids to Lagrangian particles
- 6: Calculate intermediate velocities on Lagrangian particles
- 7: Calculate intermediate locations of Lagrangian particles according to the intermediate velocities
- 8: Update link-list which manages particle distribution
- 9: Interpolate intermediate velocities from Lagrangian particles to Eulerian grids
- 10: Solve PPE
- 11: Modify intermediate velocities on Eulerian grids and Lagrangian particles
- 12: Modify particle locations according to updated velocities
- 13: Update link-list which manages particle distribution
- 14: Delete too-clustered particles/Add one particle into the empty cell/Delete particles from a cell which contains too many particles according to the value of $npcmax$
- 15: **end for**
- 16: Output results for post-processing

3.2 Cell-centered combined compact difference scheme in collocated grids

In this study, a cell-centered (CC) collocated grid system schematically shown in Fig. 1a is adopted. Red circles denote cell centers storing variables \mathbf{u} and p , green diamonds denote boundary face centers storing boundary values of \mathbf{u} and p , and blue diamonds denote corner points storing only boundary values of \mathbf{u} . The reason for keeping corner points will be addressed in Sect. 3.4.2. The numbers of the grid points and the mesh cells are defined as np and $nc = np - 1$, respectively. The grid size is defined as $h = \frac{L_x}{nc}$, where L_x is the domain length along x -direction. For the sake of convenience,

a uniform Cartesian mesh is used in this study. For spatial derivative terms in interior grid points, the sixth-order accurate combined compact difference (CCD) [58] schemes given below are applied to calculate the first and second derivative terms, respectively.

$$\begin{aligned} & \frac{7}{16} (\phi'_{i+1} + \phi'_{i-1}) + \phi'_i - \frac{h}{16} (\phi''_{i+1} - \phi''_{i-1}) \\ &= \frac{15}{16h} (\phi_{i+1} - \phi_{i-1}) \end{aligned} \tag{5}$$

$$\begin{aligned} & \frac{9}{8h} (\phi'_{i+1} - \phi'_{i-1}) - \frac{1}{8} (\phi''_{i+1} + \phi''_{i-1}) + \phi''_i \\ &= \frac{3}{h^2} (\phi_{i+1} - 2\phi_i + \phi_{i-1}) \end{aligned} \tag{6}$$

It is noted that the original CCD schemes were derived in the context of vertex-centered (VC) collocated grid system which is shown in Fig. 1b. In the cell-centered grid system, the schemes should be derived for calculating the first and second derivatives with Dirichlet and Neumann boundary conditions for boundary cells. Here the scheme for approximating the first derivative term subjected to Dirichlet boundary condition is taken as an example. The first derivative term at the first cell center, $i = 1$, is assumed to take the form of

$$\phi'_1 + a_1\phi'_2 + hb_1\phi''_2 = \frac{1}{h} (c_1\phi_L + c_2\phi_1 + c_3\phi_2) \tag{7}$$

In the above equation, h is the grid size and ϕ_L is the value at the left Dirichlet boundary. After performing the modified equation analysis and eliminating the first few leading error terms, the five coefficients shown in Eq. (7) can be determined uniquely as: $a_1 = \frac{8}{9}$, $b_1 = -\frac{1}{6}$, $c_1 = -\frac{16}{27}$, $c_2 = -1$, and $c_3 = \frac{43}{27}$. The leading error term of Eq. (7) is $\frac{1}{240}h^4\phi^{(5)}$. By repeating the above procedure, the discretization schemes for calculating the first and second derivative terms with Dirichlet, Neumann, and periodic boundary conditions can

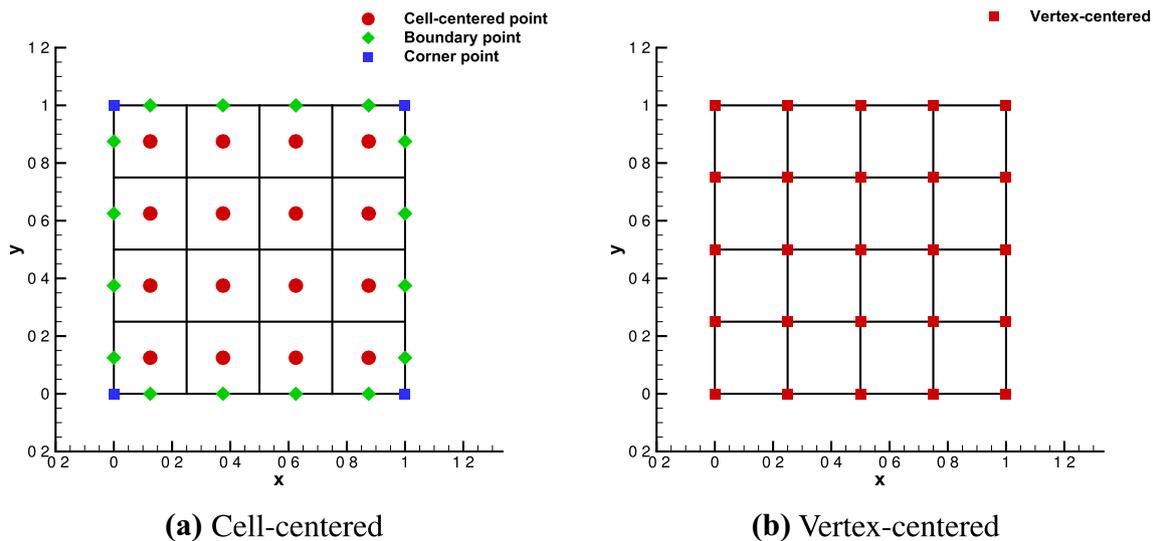


Fig. 1 Two-dimensional grid systems

be derived and the corresponding matrix equations, taking $nc = 4$ as an example, are shown below.

- (1) Cell-centered combined compact difference scheme for the case with Dirichlet boundary condition (CC-CCD-D) is

$$\left[\begin{array}{cc|cc|cc|cc} 1 & 0 & \frac{8}{9} & -\frac{h}{6} & & & & \\ 0 & 1 & \frac{2}{3h} & 0 & & & & \\ \hline \frac{7}{16} & \frac{h}{16} & 1 & 0 & \frac{7}{16} & -\frac{h}{16} & & \\ -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 & \frac{9}{8h} & -\frac{1}{8} & & \\ \hline & & \frac{7}{16} & \frac{h}{16} & 1 & 0 & \frac{7}{16} & -\frac{h}{16} \\ & & -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 & \frac{9}{8h} & -\frac{1}{8} \\ \hline & & & & \frac{8}{9} & \frac{h}{6} & 1 & 0 \\ & & & & -\frac{2}{3h} & 0 & 0 & 1 \end{array} \right] \begin{pmatrix} \phi'_1 \\ \phi''_1 \\ \phi'_2 \\ \phi''_2 \\ \phi'_3 \\ \phi''_3 \\ \phi'_4 \\ \phi''_4 \end{pmatrix} = \begin{pmatrix} -\frac{16}{27h} & -\frac{1}{h} & \frac{43}{27h} & & & & & \\ \frac{32}{9h^2} & -\frac{6}{h^2} & \frac{22}{9h^2} & & & & & \\ \hline & -\frac{15}{3h^2} & 0 & \frac{15}{3h^2} & & & & \\ & \frac{3}{h^2} & -\frac{6}{h^2} & \frac{3}{h^2} & & & & \\ \hline & & -\frac{15}{3h^2} & 0 & \frac{15}{3h^2} & & & \\ & & \frac{3}{h^2} & -\frac{6}{h^2} & \frac{3}{h^2} & & & \\ \hline & & & & -\frac{43}{22} & \frac{1}{h} & \frac{16}{32} & \\ & & & & \frac{27h}{9h^2} & -\frac{6}{h^2} & \frac{27h}{9h^2} & \end{pmatrix} \begin{pmatrix} \phi_L \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_R \end{pmatrix} \tag{8}$$

- (2) Cell-centered combined compact difference scheme for the case with Neumann boundary condition (CC-CCD-N) is

$$\left[\begin{array}{cc|cc|cc|cc} 1 & 0 & \frac{31}{27} & -\frac{2h}{9} & & & & \\ 0 & 1 & -\frac{8}{9h} & \frac{1}{3} & & & & \\ \hline \frac{7}{16} & \frac{h}{16} & 1 & 0 & \frac{7}{16} & -\frac{h}{16} & & \\ -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 & \frac{9}{8h} & -\frac{1}{8} & & \\ \hline & & \frac{7}{16} & \frac{h}{16} & 1 & 0 & \frac{7}{16} & -\frac{h}{16} \\ & & -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 & \frac{9}{8h} & -\frac{1}{8} \\ \hline & & & & \frac{31}{27} & \frac{2h}{9} & 1 & 0 \\ & & & & \frac{8}{9h} & \frac{1}{3} & 0 & 1 \end{array} \right] \begin{pmatrix} \phi'_1 \\ \phi''_1 \\ \phi'_2 \\ \phi''_2 \\ \phi'_3 \\ \phi''_3 \\ \phi'_4 \\ \phi''_4 \end{pmatrix} = \begin{pmatrix} \frac{4}{27} & -\frac{2}{h} & \frac{2}{h} & & & & & \\ \frac{8}{9h} & 0 & 0 & & & & & \\ \hline & -\frac{15}{3h^2} & 0 & \frac{15}{3h^2} & & & & \\ & \frac{3}{h^2} & -\frac{6}{h^2} & \frac{3}{h^2} & & & & \\ \hline & & -\frac{15}{3h^2} & 0 & \frac{15}{3h^2} & & & \\ & & \frac{3}{h^2} & -\frac{6}{h^2} & \frac{3}{h^2} & & & \\ \hline & & & & -\frac{2}{h} & \frac{2}{h} & \frac{4}{27} & \\ & & & & 0 & 0 & \frac{8}{9h} & \end{pmatrix} \begin{pmatrix} \phi'_L \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi'_R \end{pmatrix} \tag{9}$$

- (3) Cell-centered combined compact difference scheme for the case with periodic boundary condition (CC-CCD-P) is

$$\left[\begin{array}{cc|cc|cc|cc} 1 & 0 & \frac{7}{16} & -\frac{h}{16} & & \frac{7}{16} & \frac{h}{16} & \\ 0 & 1 & \frac{9}{8h} & -\frac{1}{8} & & -\frac{9}{8h} & -\frac{1}{8} & \\ \hline \frac{7}{16} & \frac{h}{16} & 1 & 0 & \frac{7}{16} & -\frac{h}{16} & & \\ -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 & \frac{9}{8h} & -\frac{1}{8} & & \\ \hline & & \frac{7}{16} & \frac{h}{16} & 1 & 0 & \frac{7}{16} & -\frac{h}{16} \\ & & -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 & \frac{9}{8h} & -\frac{1}{8} \\ \hline \frac{7}{16} & -\frac{h}{16} & & & \frac{7}{16} & \frac{h}{16} & 1 & 0 \\ \frac{9}{8h} & -\frac{1}{8} & & & -\frac{9}{8h} & -\frac{1}{8} & 0 & 1 \end{array} \right] \begin{pmatrix} \phi'_1 \\ \phi''_1 \\ \phi'_2 \\ \phi''_2 \\ \phi'_3 \\ \phi''_3 \\ \phi'_4 \\ \phi''_4 \end{pmatrix} = \begin{pmatrix} 0 & \frac{15}{16h} & -\frac{15}{3h} & & & & & \\ -\frac{6}{h^2} & \frac{15}{3h^2} & \frac{15}{3h^2} & & & & & \\ \hline -\frac{15}{16h} & 0 & \frac{15}{3h^2} & & & & & \\ \frac{3}{h^2} & -\frac{6}{h^2} & \frac{3}{h^2} & & & & & \\ \hline -\frac{15}{16h} & 0 & \frac{15}{3h^2} & & & & & \\ \frac{3}{h^2} & -\frac{6}{h^2} & \frac{3}{h^2} & & & & & \\ \hline \frac{15}{16h} & -\frac{15}{16h} & 0 & & & & & \\ \frac{3}{h^2} & \frac{3}{h^2} & -\frac{6}{h^2} & & & & & \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} \tag{10}$$

tridiagonal matrices. Note that the size of the sub-matrix is two because the number of unknowns at each mesh cell is two and the number of sub-matrices is nc . Advantages of adopting CC grids rather than VC grids are listed below.

1. There are no geometrical singular points. As shown in Fig. 1b, the four corner points are the geometrically singular points upon which numerical implementation of

Neumann boundary conditions is not easy while solving PPE. The Neumann boundary conditions can be discretized along the diagonal grid points as stated in [59]

As shown in Eqs. (8) and (9), the coefficient matrices for the calculation of the first and second derivative terms are block-

for uniform mesh or using the modified Collatz difference scheme given in [60]. However, one can avoid discretiz-

ing the Neumann boundary conditions at the corner points in CC grids.

2. The matrix size, LDA, is constant for the cases with Dirichlet, Neumann, and periodic boundary conditions, while LDA is different under different boundary conditions if VC grid is adopted. For example, $LDA = 2 \times nc$ for calculating the first and second derivative values with the given function values and $LDA = 3 \times nc$ for solving partial differential equation (PDE) if CC-CCD scheme is adopted. On the other hand, LDA equals $2 \times np$ and $2 \times nc$ for VC-CCD scheme to deal with problems with Dirichlet and Neumann/periodic boundary conditions. Therefore, the complexity in programming can be reduced in using the CC-CCD scheme.
3. In the original VC-CCD scheme, three function values at grid points $i = 1-3$ are taken into account in calculating the derivative terms at grid point $i = 1$. The derived block-tridiagonal matrix property is broken when solving a PDE as shown below.

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{C}_1 & & & & \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & & & \\ & \mathbf{A}_3 & \mathbf{B}_3 & \mathbf{C}_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_{nc-1} & \mathbf{B}_{nc-1} & \mathbf{C}_{nc-1} \\ & & & & & \mathbf{A}_{nc} & \mathbf{B}_{nc} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \vdots \\ \mathbf{x}_{nc-1} \\ \mathbf{x}_{nc} \end{Bmatrix} = \begin{Bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_{nc-1} \\ \mathbf{b}_{nc} \end{Bmatrix} \tag{12}$$

In the above matrix equation, \mathbf{A}_i , \mathbf{B}_i , and \mathbf{C}_i are the sub-matrices shown in the coefficient matrix and their dimension is 2 for calculating the first and second derivative terms and 3 for solving a PDE. \mathbf{x}_i and \mathbf{b}_i are the solution vector and source vector corresponding to the cell index i . The coefficient matrix is firstly decomposed into \mathbf{L} and \mathbf{U} matrices as shown below.

$$\left[\begin{array}{ccc|ccc|} a & b & c & 0 & 0 & 0 & & & \\ \alpha & \beta & 0 & 0 & 0 & 0 & & & \\ \frac{31}{h} & 14 & 2h & -\frac{32}{h} & 16 & -4h & \frac{1}{h} & & \\ \hline 0 & 0 & 0 & a & b & c & 0 & 0 & 0 \\ \frac{15}{16h} & \frac{7}{16} & \frac{h}{16} & 0 & 1 & 0 & -\frac{15}{16h} & \frac{7}{16} & -\frac{h}{16} \\ -\frac{3}{h^2} & -\frac{9}{8h} & -\frac{1}{8} & \frac{6}{h^2} & 0 & 1 & -\frac{3}{h^2} & \frac{9}{8h} & -\frac{1}{8} \\ \hline & & & 0 & 0 & 0 & a & b & c \\ & & & \frac{15}{16h} & \frac{7}{16} & \frac{h}{16} & 0 & 1 & 0 \\ & & & -\frac{3}{h^2} & -\frac{9}{8h} & -\frac{1}{8} & \frac{6}{h^2} & 0 & 1 \\ \hline & & & 0 & 0 & 0 & a & b & c \\ & & & \frac{15}{16h} & \frac{7}{16} & \frac{h}{16} & 0 & 1 & 0 \\ & & & -\frac{3}{h^2} & -\frac{9}{8h} & -\frac{1}{8} & \frac{6}{h^2} & 0 & 1 \\ \hline & & & 0 & 0 & 0 & a & b & c \\ & & & 0 & 0 & 0 & \alpha & \beta & 0 \\ & & & -\frac{1}{h} & \frac{32}{h} & 16 & 4h & -\frac{31}{h} & 14 & -2h \end{array} \right] \begin{Bmatrix} \phi_1 \\ \phi_1' \\ \phi_1'' \\ \phi_2 \\ \phi_2' \\ \phi_2'' \\ \phi_3 \\ \phi_3' \\ \phi_3'' \\ \phi_4 \\ \phi_4' \\ \phi_4'' \\ \phi_5 \\ \phi_5' \\ \phi_5'' \end{Bmatrix} = \begin{Bmatrix} src_1 \\ \gamma \\ 0 \\ src_2 \\ 0 \\ 0 \\ src_3 \\ 0 \\ 0 \\ src_4 \\ 0 \\ 0 \\ src_5 \\ \gamma \\ 0 \end{Bmatrix} \tag{11}$$

In the above matrix equation, coefficients a , b , and c are used in the governing equation, which is $a\phi + b\phi' + c\phi'' = src$, and α , β , and γ are shown in Eq. (4). From the computational point of view, the currently adopted CC-CCD scheme retains the block-tridiagonal matrix property while calculating derivative values and solving a PDE. Therefore, the block-tridiagonal LU decomposition method can be adopted in this paper for solving Eqs. (8) and (9).

LU decomposition of the block-tridiagonal matrix is illustrated below. Equations (8) and (9) can be written in the following form.

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{C}_1 & & & & \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & & & \\ & \mathbf{A}_3 & \mathbf{B}_3 & \mathbf{C}_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_{nc-1} & \mathbf{B}_{nc-1} & \mathbf{C}_{nc-1} \\ & & & & & \mathbf{A}_{nc} & \mathbf{B}_{nc} \end{bmatrix}$$

MLS interpolation method. The subscript i is the index of the interpolated point (blue solid triangle), and r is the distance between the interpolated point and the reference point (red solid circle). In this study, the weighting function given below takes the form from [62].

$$f(x, y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + O(x^3, y^3) \tag{19}$$

The coefficients $a_0 - a_5$ can be determined by using the MLS method to minimize the error and solving the following linear system.

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

$$\omega_i(r_i, r_e) = \begin{cases} \frac{a}{r_e} + \frac{b}{r_e} \left(\frac{r_i}{r_e}\right) + \frac{c}{r_e} \left(\frac{r_i}{r_e}\right)^2 + \frac{d}{r_e} \left(\frac{r_i}{r_e}\right)^3 + \frac{e}{r_e} \left(\frac{r_i}{r_e}\right)^4 & 0 \leq r_i \leq r_e \\ 0 & r_e > r_i \end{cases} \tag{18}$$

where

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

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

For performing the MLS interpolation method, a local polynomial given below is constructed.

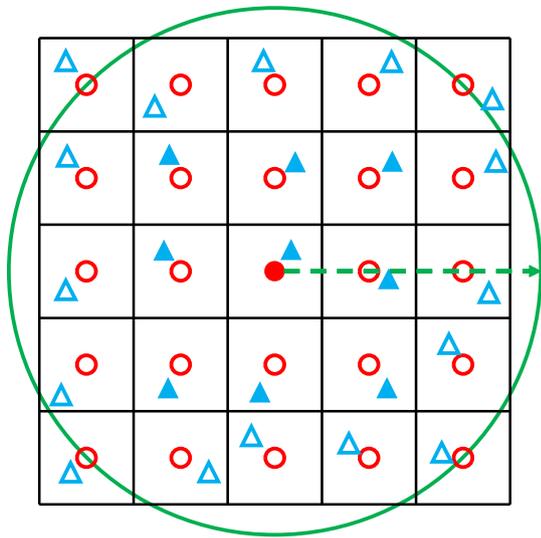


Fig. 2 Schematic of the interpolation process (circles denote Eulerian grids; triangles denote Lagrangian particles; solid symbols denote the points participating in interpolation process; hollow symbols denote the points not participating in interpolation process)

It is worth noting that the function value can be evaluated at the reference point by calculating $f(0, 0)$ which is actually a_0 . Therefore, the above linear system does not need to be solved entirely to get all the six coefficients. Instead, backward elimination procedure can be adopted to get a_0 without computing the values of $a_1 - a_5$ and it is more computationally efficient.

Since the interpolated intermediate velocity field is not divergence free, a modified step is introduced to obtain the divergence free velocity field.

$$\mathbf{u}^{n+1} = \mathbf{u}_{\text{interpolated}}^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \tag{21}$$

Taking divergence operation on the above equation and using Eq. (1), the pressure Poisson equation (PPE) can be obtained as following.

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}_{\text{interpolated}}^* \tag{22}$$

In this study, the above PPE is discretized by the second-order central difference scheme, leading to

$$p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j} = \frac{\rho h}{2\Delta t} (u_{i+1,j}^* - u_{i-1,j}^* + v_{i,j+1}^* - v_{i,j-1}^*) \tag{23}$$

The resulting linear system is solved by the conjugate gradient (CG) iterative solver. The iterative solution of p is said to be convergent if the root mean square (RMS) of the residuals is smaller than a convergence criteria, said RMS (res^k) $< \varepsilon = 10^{-15}$, where res^k is the residual of the k -th step iterative solutions and the function RMS is defined as

$$\text{RMS}(\phi) = \left(\frac{1}{nc_x} \frac{1}{nc_y} \sum_{i=1}^{nc_x} \sum_{j=1}^{nc_y} \phi_{i,j}^2 \right)^{\frac{1}{2}} \tag{24}$$

In the above equation, nc_x and nc_y are the number of mesh cells in x - and y -directions, respectively.

For the sake of completeness, the numerical procedure for the IMLE method is summarized below.

Algorithm 5 Algorithm of the IMLE method

- 1: Read input file
 - 2: Calculate computational parameters
 - 3: **for** $time < MaxTime$ **do**
 - 4: Calculate the diffusion terms on Eulerian grids, Eq. (8) or (9) or (10).
 - 5: Calculate the intermediate velocities on Eulerian grids, Eq. (16).
 - 6: Calculate the intermediate locations of the Lagrangian particles departing from the Eulerian grids according to the intermediate velocities, Eq. (17).
 - 7: Interpolate the intermediate velocities from Lagrangian particles to Eulerian grids, Eqs. (20).
 - 8: Solve the PPE, Eq. (23).
 - 9: Update the intermediate velocities on Eulerian grids, Eq. (21).
 - 10: **end for**
 - 11: Output results for post-processing
-

Some key differences between the MLE and IMLE methods are summarized below.

1. After the interpolation step (step 9 in Algorithm 1), MLE particles keep moving from the intermediate locations \mathbf{r}^* to the next locations \mathbf{r}^{n+1} (step 12 in Algorithm 1) until they are deleted. This means that MLE particles are not always located at the Eulerian grid points. Thus another interpolation (step 5 in Algorithm 1) is required to interpolate diffusion terms from Eulerian grids to Lagrangian particles after those terms have been calculated on Eulerian grids by the VC-CCD scheme. Therefore, the accuracy order of the diffusion terms on Lagrangian particles is restricted by the accuracy order of the adopted interpolation scheme.
2. In every advection step (step 7 in Algorithm 1 and step 6 in Algorithm 5), the departing points of the MLE particles may not be located on the Eulerian grid points, while the IMLE particles are exactly departed from the Eulerian grid points (shown in Fig. 3). In other words, distance between a Lagrangian particle and its corresponding Eulerian grid point may be longer for MLE method, i.e., $|\mathbf{r}^n - \mathbf{r}^{n-2}|_{MLE} > |\mathbf{r}^* - \mathbf{r}^n|_{IMLE}$. As a result, IMLE particles can be expected to be distributed more uniformly than MLE particles (a small/large deviation from the uniform Eulerian grids in IMLE/MLE methods). In this situation, particle distribution with smaller deviation can reduce MLS interpolation error [63] in IMLE method.
3. The source term of the PPE is calculated from the interpolated velocities. It is therefore better to update the

interpolated velocities on Eulerian grids only rather than to update the velocities on Lagrangian particles as well.

4. The interpolation procedure (step 7 in Algorithm 5) can be seen as a re-meshing step which re-meshes the slightly non-uniform Lagrangian particles to the uniform Eulerian grids.
5. MLE uses semi-staggered grid which requires one (two) additional interpolation procedures to interpolate the pressure gradient terms from cell faces to cell vertices in two (three)-dimensional grids [64]. IMLE adopts collocated grid, and there is no need for interpolating pressure gradient term; therefore, IMLE is more efficient.

With respect to the conventional finite difference method, which directly discretizes the convection terms, the previous process (steps 6–7 in Algorithm 5) can be seen as an alternative way to compute the addition of particle derivative and convection terms at grid points. As shown above, the particles move along the streamlines so that there is no false diffusion error generated from using any local one-dimensional upwinding scheme. Also, since there is no convection terms used in IMLE method, there is no dispersion error as well resulting from the discretization of the convection terms.

3.4 Boundary treatments for intermediate velocity

3.4.1 Boundary treatment for Neumann boundary condition

For the cases of Dirichlet boundary condition, intermediate velocity components located on boundary face centers (green diamonds in Fig. 1a) are directly assigned as velocity components at next time step, i.e., $\mathbf{u}^* = \mathbf{u}^{n+1}$. On the other hand, if Neumann boundary condition is considered, velocity components and tangential location of a boundary face particle are determined by the boundary cell particle. As shown in Fig. 4, tangential location of a boundary face particle is the projection of the location of a boundary cell particle, i.e., $y_{\text{face particle}}^* = y_{\text{cell particle}}^*$. Similarly, velocity vector is duplicated from the boundary cell particle to its corresponding boundary face particle, i.e., $\mathbf{u}_{\text{face particle}}^* = \mathbf{u}_{\text{cell particle}}^*$.

It is worth noting that there are no ghost/dummy particles adopted in IMLE method, while they are usually used in conventional SPH [25,65,66] and MPS [8,9,67] methods. Ghost particle or repulsive force model, which may deteriorate computational accuracy, was developed to prevent particles from penetrating rigid walls. However, in the proposed IMLE method, Lagrangian particles are reset to uniform Eulerian mesh points, thus ensuring that no particle can penetrate across the walls.

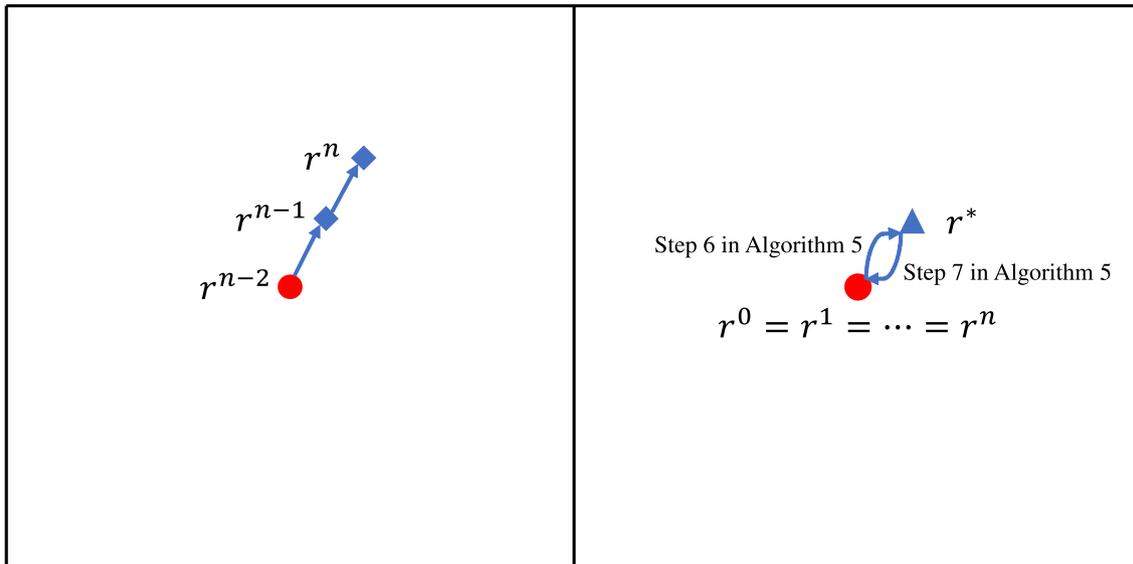


Fig. 3 Schematic of MLE (left) and IMLE (right) particles. Red circles denote cell centers, blue diamonds denote MLE particles at different time step, and blue triangle denotes intermediate location of IMLE particle. (Color figure online)

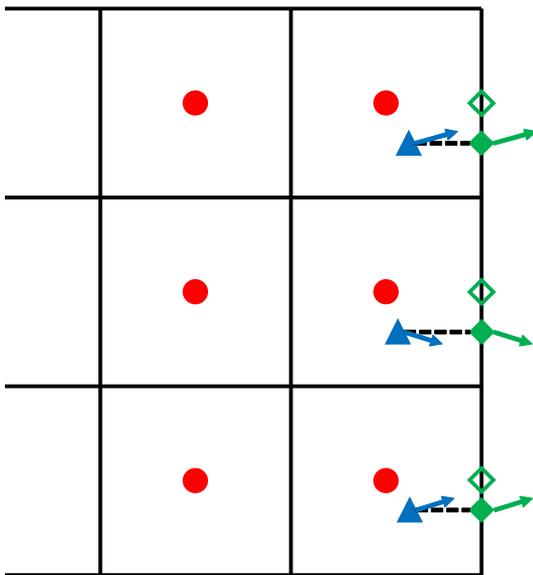


Fig. 4 Schematic of determining location and velocity of boundary face particles (green solid diamond) for Neumann boundary condition in x -direction. (Color figure online)

3.4.2 Boundary treatment for interpolation procedure

When using MLS interpolation, there are six unknowns in Eqs. (19) and (20). It is obvious that there should be at least six particles needed to be taken into account for MLS interpolation. Also, MLS interpolation is more accurate if there are more particles when solving Eq. (20). In this study, nine ($= 3^d$, where $d = 2$ is the problem dimension) particles are considered in MLS interpolation for interior, boundary and

corner cells. Three interpolation cases are shown in Figs. 2 and 5a, b. In order to maintain nine-particle property, four corner particles are reserved for utilizing MLS interpolation of intermediate velocities. As mentioned previously, solid triangles and diamonds are used to interpolate intermediate velocities to solid circle while not for hollow symbols.

A most important point in using MLS interpolation is that there is no need to tackle the difficulty of the unity (normalization) condition which is the inevitable condition for SPH kernel function. In SPH, a physical quantity ϕ on a particle can be calculated in continuous sense by means of

$$\phi_i = \int \phi_j W(\mathbf{x}_j - \mathbf{x}_i, h) dV_j \quad (25)$$

The discrete form of the above equation is

$$\phi_i = \sum_{j=1}^{NP} \phi_j W(\mathbf{x}_j - \mathbf{x}_i, h) \Delta V_j \quad (26)$$

In the above two equations, the quantity ϕ is interpolated from the surrounding particles $j = 1 \sim NP$ to the particle i , where NP is the number of particles within the influence domain of particle i , W the kernel function, h the smoothing length, and ΔV_j the volume of particle j . One of the several important properties for W is that W should satisfy the unity (normalization) condition. Continuous and discrete forms of the unity condition are shown below.

$$\int W(\mathbf{x}_j - \mathbf{x}_i, h) dV_j = 1 \quad (27)$$

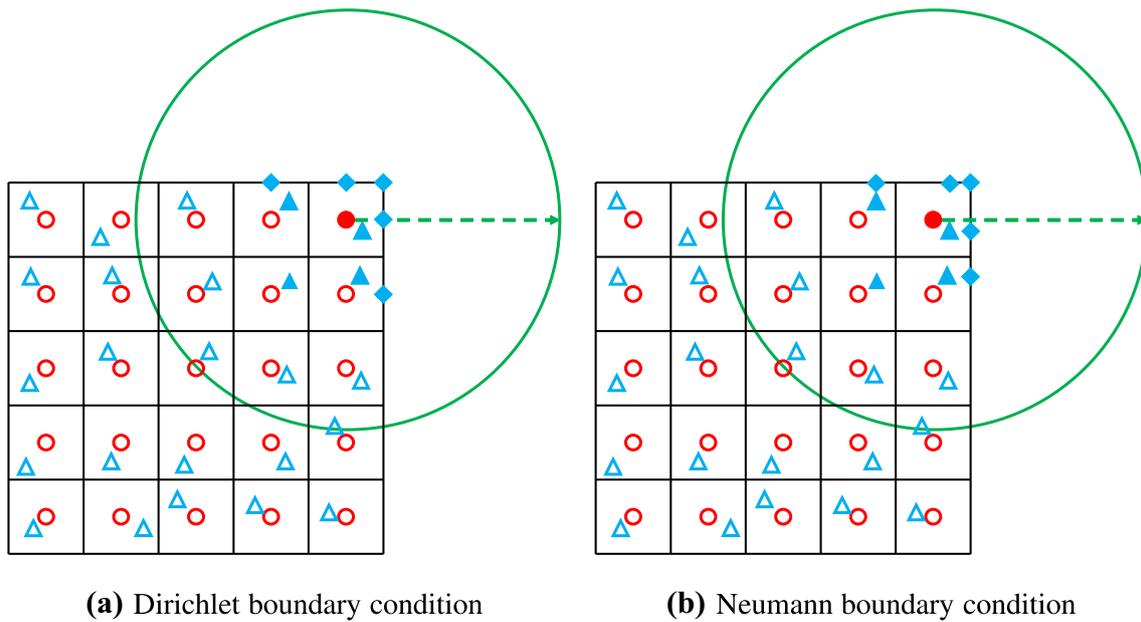


Fig. 5 Schematic of interpolation procedure for different boundary conditions for a corner cell

Table 1 Spatial rates of convergence for the Taylor–Green flow

Mesh size	u		v		p	
	RMS of error	sroc	RMS of error	sroc	RMS of error	sroc
1.00E-1	2.6800E-03	–	2.6800E-03	–	9.6307E-02	–
5.00E-2	1.7601E-04	3.9285	1.7601E-04	3.9285	2.2690E-02	2.0856
2.50E-2	1.1140E-05	3.9818	1.1147E-05	3.9809	4.1758E-03	2.4419
1.25E-2	7.0159E-07	3.9890	7.6250E-07	3.8698	1.2609E-03	1.7276

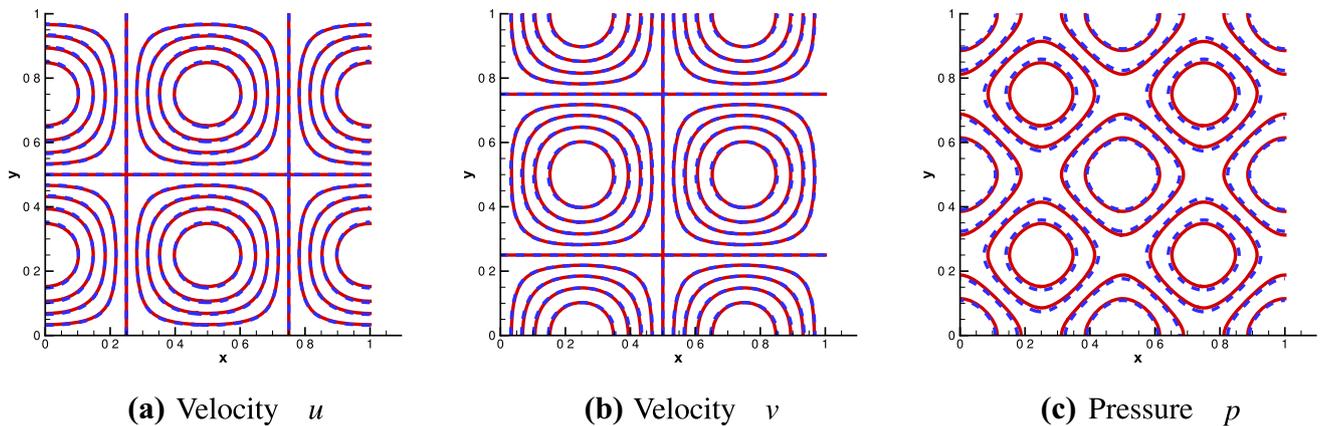


Fig. 6 Numerical results of the Taylor–Green vortex flow considered at $Re = 10^6$

$$\sum_{j=1}^{NP} W(\mathbf{x}_j - \mathbf{x}_i, h) \Delta V_j = 1 \tag{28}$$

However, most of the kernel functions were developed in continuous sense, i.e., satisfying Eq. (27) only, they are not guaranteed to satisfy the discrete unity condition as stated

in [68–70]. In order to enforce the unsatisfied discrete unity condition, reproducing kernel particle method (RKPM) [71] or improved SPH methods [69,70] were developed.

On the other hand, MLS interpolation method is based on the concept of surface reconstruction. By using Eq. (19), a smooth function can be reconstructed accurately under the nine-particle framework. In a short summary, under the

Fig. 7 Evolution of decayed maximum velocity

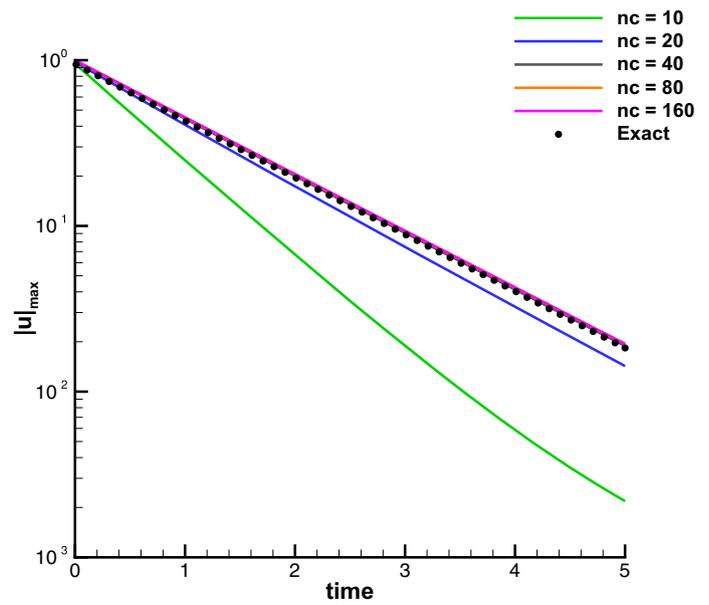


Fig. 8 The maximum of the relative errors obtained for the cases with different grid numbers

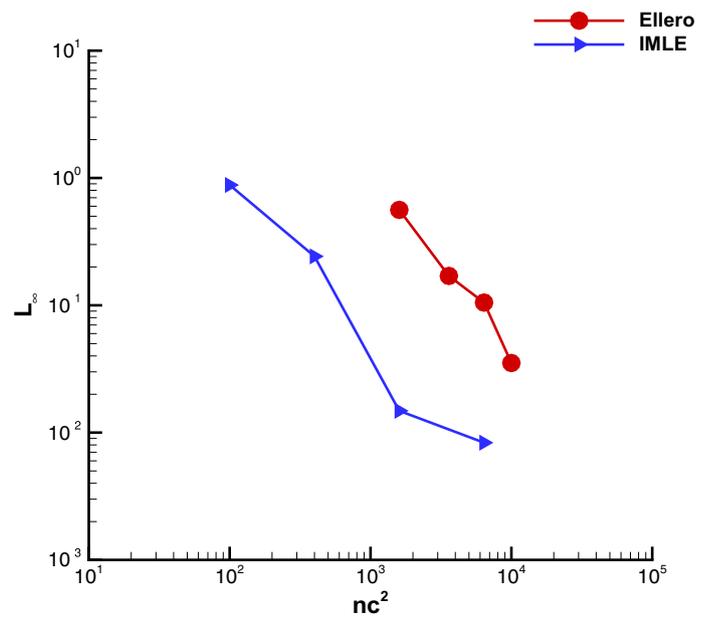
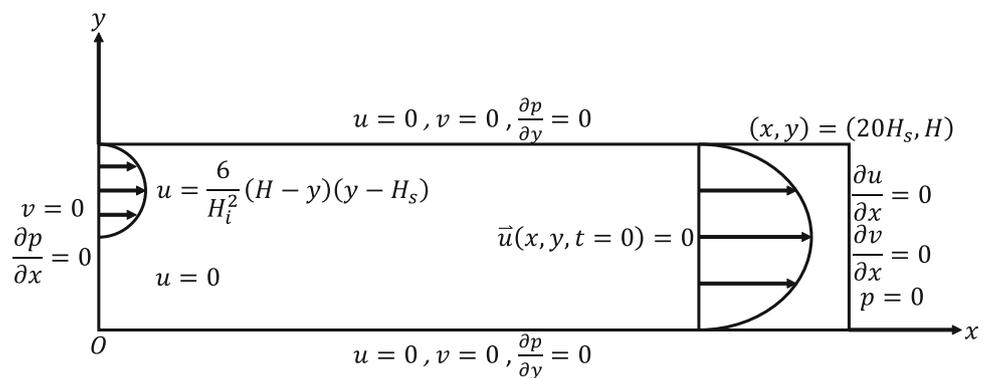


Fig. 9 Schematic of the backward-facing step flow problem



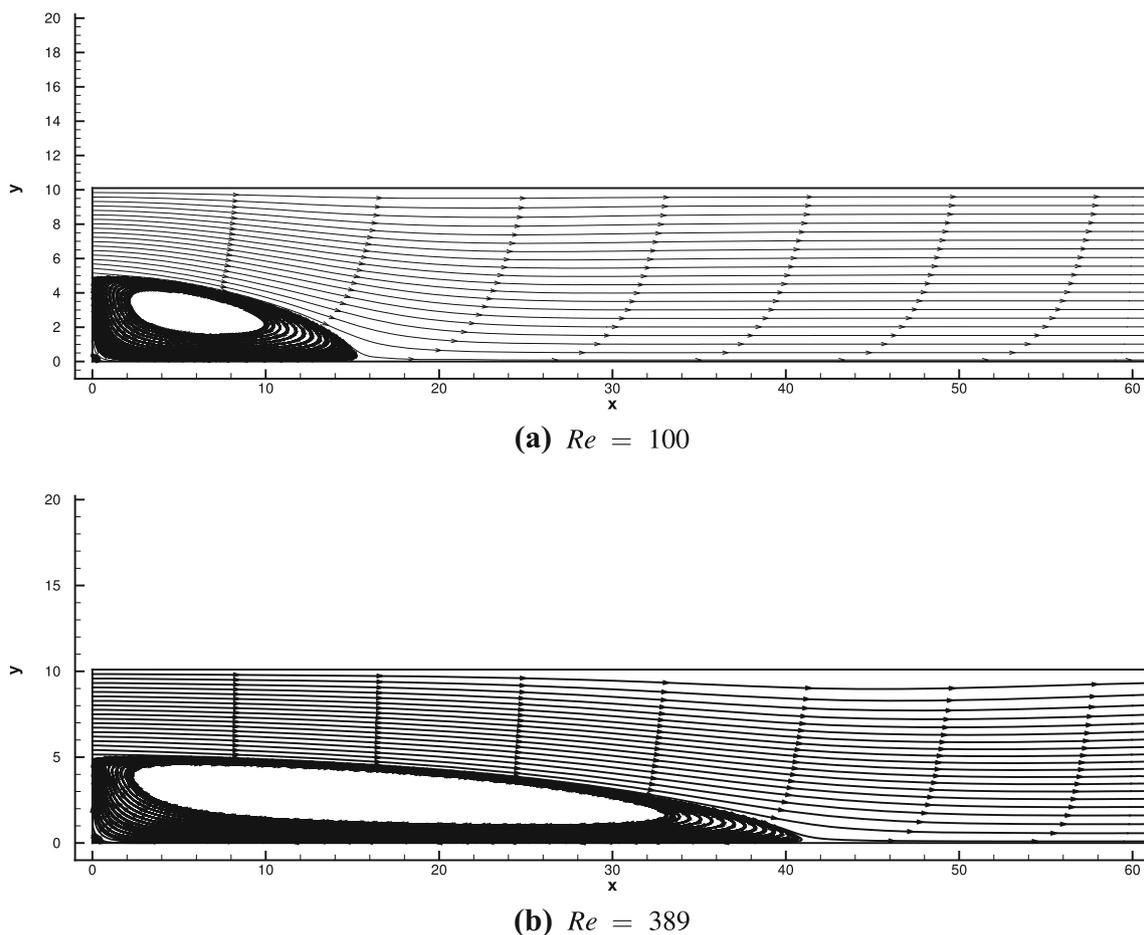


Fig. 10 Streamlines for different Reynolds numbers of the backward-facing step flow problem

weakly non-uniform particle distribution (stated in Sect. 3.3), the nine-particle compact framework (shown in this section) adopted in MLS interpolation method can improve computational accuracy.

4 Verification and validation studies

4.1 Taylor–Green vortex flow

In the first test case, the IMLE method is verified using the analytical solutions of the Taylor–Green vortex flow problem. The considered flow domain is a unit square consisting of a series of decaying vortices. The analytical solutions of the flow variables can be expressed as follows.

$$u_{\text{analytic}}(x, y, t) = -Ue^{bt} \cos(2\pi x) \sin(2\pi y) \tag{29}$$

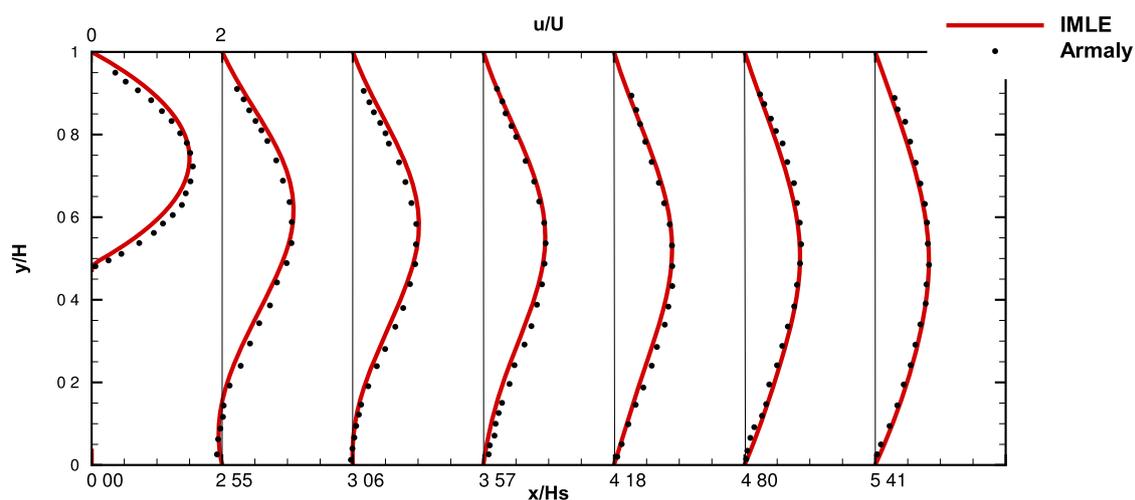
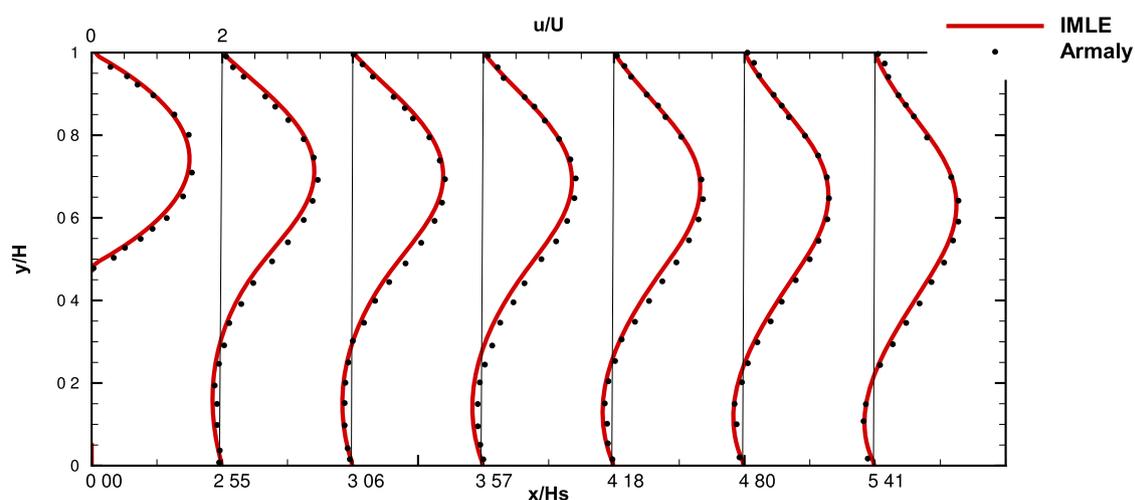
$$v_{\text{analytic}}(x, y, t) = Ue^{bt} \sin(2\pi x) \cos(2\pi y) \tag{30}$$

$$p_{\text{analytic}}(x, y, t) = -\frac{U^2}{4}e^{2bt} [\cos(4\pi x) + \cos(4\pi y)] \tag{31}$$

In the above equations, b is the decaying rate expressed as $-8\pi^2/Re$, where $Re (= \rho UL/\mu)$ is the Reynolds number ($= 100$). The density ρ , reference velocity U , and characteristic length L are set to unity similar to those reported in the previous works [24,55,72].

First of all, in order to test the spatial rates of convergence (sroc) of velocity and pressure, periodic boundary conditions are applied in both x - and y -directions. Different grid resolutions ($nc^2 = 10^2, 20^2, 40^2$, and 80^2) are adopted. In order to evaluate the sroc, a relatively small time step size ($\Delta t = 10^{-5}$) is considered and the simulations are executed for only one time step in order to eliminate the influence of time discretization error on the computed sroc [73]. Table 1 shows the RMS of the errors and the sroc at different grid resolutions. It is observed that velocity and pressure exhibit fourth- and second-order accuracy, respectively. Therefore, the current IMLE method is more accurate than our previous MLE method [43], as the latter reported that the sroc values of velocity and pressure are merely 2.0 and 1.4, respectively.

Secondly, in order to examine the numerical dissipation of the proposed IMLE method, the Reynolds number is


 (a) $Re = 100$

 (b) $Re = 389$
Fig. 11 Normalized velocity profiles of the backward-facing step flow problem

increased to 10^6 . The grid resolution is set to $nc^2 = 400^2$ in this case, time step size $\Delta t = 0.1\Delta x$ is considered such that the initial Courant–Friedrichs–Lewy (CFL) number is about 0.1. Figure 6 shows the contours of velocities u , v , and pressure p . It is appealing to note that the predicted numerical solutions vary smooth even at high Re flow condition.

Finally, the proposed IMLE solutions are compared with the ISPH solutions [72]. Here, the time evolutions of maximum velocity (Eq. 32) are compared. The maximum relative error is defined in Eq. (33).

$$u_{\max}(t) = \max(\text{abs}(\mathbf{u}(t))) \quad (32)$$

$$L_{\infty} = \max_{t=0}^5 \frac{|u_{\max}(t) - e^{bt}|}{e^{bt}} \quad (33)$$

For this comparative study, simulations are performed using grid resolutions $nc^2 = 10^2, 20^2, 40^2$, and 80^2 and the CFL number is again set as 0.1. The Reynolds number is 100, and the simulation time is 5 s. Figure 7 shows that $u_{\max}(t)$ comes closer to the analytical solution if a finer mesh is adopted. Figure 8 shows that the proposed IMLE method outperforms the ISPH method [72].

4.2 Backward-facing step flow

In this subsection, a backward-facing step flow problem is considered to show the ability of applying IMLE to simulate inflow-outflow problem. The physical domain, the initial, and the boundary conditions are shown in Fig. 9. In this

Table 2 Comparisons of computation times between MLE and IMLE of the backward-facing step flow problem at $Re = 100$

Program name	MLE		IMLE		
	Comput. time (s)	Prop. (%)	Comput. time (s)	Prop. (%)	Improvement
Main	1.2561E+05	100.00	3.1721E+04	100.00	3.96
\mathbf{u}^*	1.1621E+03	0.93	6.1763E+02	1.95	1.88
MLS intrpl.	2.7307E+04	21.74	2.0338E+03	6.41	13.43
p^{n+1}	9.4545E+04	75.27	2.8761E+04	90.67	3.29

Table 3 Velocity u at different cross sections of the backward-facing step flow problem at $Re = 100$

y	x/H_s						
	0.00	2.55	3.06	3.57	4.18	4.80	5.41
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.5	0.0000	-0.0492	0.0006	0.0517	0.0996	0.1277	0.1407
1.0	0.0000	-0.0500	0.0341	0.1179	0.1966	0.2444	0.2673
1.5	0.0000	-0.0064	0.0993	0.2005	0.2946	0.3525	0.3814
2.0	0.0000	0.0760	0.1932	0.2996	0.3958	0.4546	0.4842
2.5	0.0000	0.1903	0.3106	0.4126	0.5001	0.5515	0.5767
3.0	0.0000	0.3288	0.4442	0.5338	0.6042	0.6419	0.6583
3.5	0.0000	0.4826	0.5846	0.6549	0.7020	0.7219	0.7269
4.0	0.0000	0.6420	0.7215	0.7660	0.7861	0.7866	0.7795
4.5	0.0000	0.7955	0.8434	0.8574	0.8487	0.8305	0.8127
5.0	0.1151	0.9299	0.9387	0.9197	0.8834	0.8493	0.8236
5.5	0.6115	1.0310	0.9970	0.9458	0.8861	0.8408	0.8109
6.0	1.0001	1.0856	1.0103	0.9320	0.8557	0.8050	0.7752
6.5	1.2775	1.0846	0.9751	0.8783	0.7943	0.7443	0.7185
7.0	1.4439	1.0249	0.8937	0.7895	0.7070	0.6631	0.6444
7.5	1.4993	0.9110	0.7737	0.6734	0.6007	0.5671	0.5570
8.0	1.4439	0.7542	0.6268	0.5404	0.4833	0.4617	0.4602
8.5	1.2774	0.5710	0.4667	0.4009	0.3620	0.3519	0.3572
9.0	1.0001	0.3793	0.3066	0.2640	0.2423	0.2409	0.2497
9.5	0.6117	0.1953	0.1568	0.1361	0.1279	0.1306	0.1385
10.1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

problem, the heights of step and inlet channel are set at $H_s = 4.9$ and $H_i = 5.2$, respectively. For simplicity, a fully developed parabolic inflow profile is prescribed at the inlet channel. The length of the physical domain is twenty times the step height. At outflow boundary, Neumann and Dirichlet boundary conditions for velocity and pressure are specified. Reynolds number for this problem is defined as $Re = \rho UL/\nu$, where ρ is the density which is 1, U the average inflow velocity which is 1 and L the hydraulic length which is equal to $2H_i$. Two different Reynolds numbers are considered in this study which are 100 and 389. Uniform grid size $\Delta x = \Delta y = 0.1$ and time step $\Delta t = 0.01$ are adopted so that Courant number is approximately equal to 0.15. Simulation times of $T = 500.0$ and 1000.0 are taken for the cases with low and high Reynolds numbers, respectively.

In order to compare the numerical results with the experimental data [74], normalized velocity profiles (u/U) at different cross sections (x/H_s) are compared. First of all, streamlines near the inlet channel are shown in Fig. 10. IMLE results show very smooth streamlines of the primary vortices. Secondly, Fig. 11 shows that velocity profiles obtained from IMLE method are in good agreement with the reference experimental data. Finally, Table 2 shows that IMLE outperforms MLE in terms of the computational efficiency. Some discussions about efficiency are detailed in the next section. The degree of computational improvement is defined as the fraction of computation time of MLE method over that of IMLE method. The numerical data of velocity u at different cross sections are shown in Tables 3 and 4 for different Reynolds numbers.

Table 4 Velocity u at different cross sections of the backward-facing step flow problem at $Re = 389$

y	x/H_s							
	0.00	2.55	3.06	3.57	4.18	4.80	5.41	
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.5	0.0000	-0.0749	-0.0865	-0.0967	-0.1072	-0.1150	-0.1183	
1.0	0.0000	-0.1251	-0.1409	-0.1536	-0.1643	-0.1685	-0.1629	
1.5	0.0000	-0.1475	-0.1606	-0.1685	-0.1705	-0.1621	-0.1397	
2.0	0.0000	-0.1399	-0.1443	-0.1419	-0.1290	-0.1028	-0.0602	
2.5	0.0000	-0.1014	-0.0927	-0.0764	-0.0455	-0.0000	0.0622	
3.0	0.0000	-0.0314	-0.0069	0.0248	0.0742	0.1377	0.2165	
3.5	0.0000	0.0721	0.1132	0.1603	0.2264	0.3045	0.3948	
4.0	0.0000	0.2140	0.2701	0.3300	0.4085	0.4953	0.5896	
4.5	0.0000	0.4003	0.4662	0.5329	0.6156	0.7018	0.7896	
5.0	0.1139	0.6311	0.6966	0.7601	0.8352	0.9086	0.9773	
5.5	0.6118	0.8882	0.9403	0.9890	1.0434	1.0916	1.1296	
6.0	1.0001	1.1306	1.1600	1.1854	1.2094	1.2238	1.2239	
6.5	1.2775	1.3119	1.3161	1.3152	1.3056	1.2833	1.2437	
7.0	1.4439	1.4023	1.3829	1.3569	1.3149	1.2580	1.1827	
7.5	1.4994	1.3907	1.3503	1.3021	1.2324	1.1472	1.0451	
8.0	1.4439	1.2754	1.2177	1.1527	1.0636	0.9614	0.8468	
8.5	1.2775	1.0609	0.9935	0.9209	0.8264	0.7235	0.6142	
9.0	1.0001	0.7635	0.6996	0.6339	0.5519	0.4664	0.3801	
9.5	0.6118	0.4176	0.3742	0.3310	0.2787	0.2262	0.1756	
10.1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

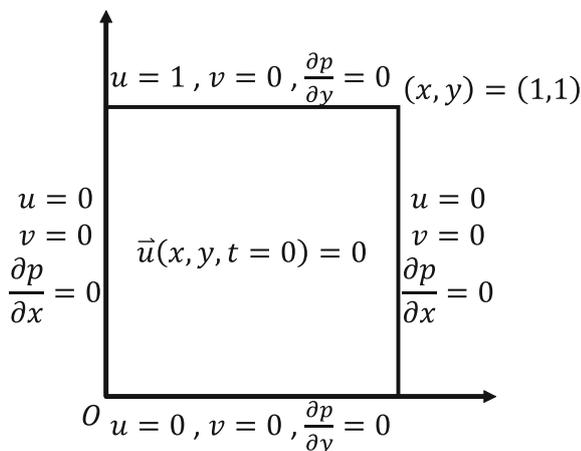


Fig. 12 Schematic of the lid-driven cavity flow problem

4.3 Lid-driven cavity flow

In the last validation study, a lid-driven cavity flow in a unit square domain is considered. The physical domain, the initial, and the boundary conditions are schematically shown in Fig. 12. For this problem, characteristic velocity and length and density are 1. Reynolds numbers of 1000 and 10,000 are considered, while the corresponding simulation times are 400.0 and 1000.0. Grid numbers 81^2 and 201^2 are adopted for

Table 5 Grid settings for MLE simulations of the lid-driven cavity flow problem

	Re	
	1000	10,000
	41-1(1)	81-1(4)
*	41-4(2)	81-4(5)
**	81-1(3)	201-1(6)

low and high Reynolds numbers, respectively. For comparisons, grid settings for MLE simulations are listed in Table 5, where the notation $m - n$ means that m is the number of grid points and n is the maximum number of particles in a cell. The row with symbol * indicates that the numerical results of those settings are shown in [43]. The row with symbol ** indicates that under these settings, particle numbers are the same for both MLE and IMLE methods. Case numberings are shown in parentheses.

Usually, velocity profiles $u(y)$ and $v(x)$ at the cross sections $x = 0.5$ and $y = 0.5$ are compared with the reference data obtained from [75] for benchmarking purpose. As shown in Fig. 13, under the same number of grids/particles ($nc^2 = 80^2, 200^2$), IMLE method shows better accuracy than that of MLE(3)/MLE(6). Both high accuracy order of approximating diffusion terms and smaller MLS interpolation error can result in this result. On the other hand, for the previously proposed MLE method, larger values of n

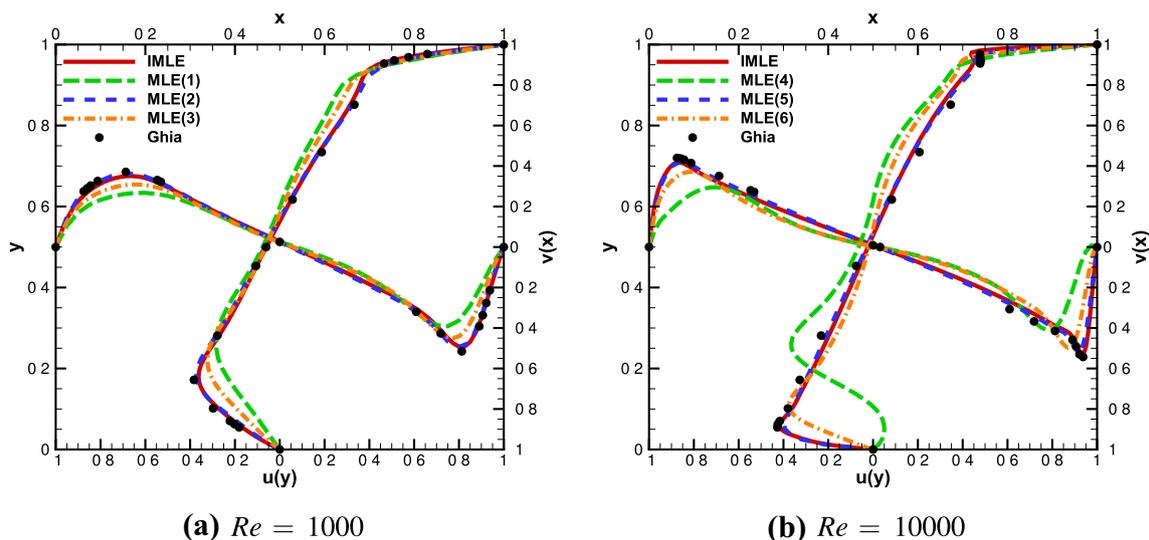


Fig. 13 Comparison of the velocity profiles between IMLE, MLE and Ghia [75] for different Reynolds numbers of the lid-driven cavity flow problem

(more particles) with the same number of grid points show more accurate velocity profiles (comparing cases (1)–(2) and (4)–(5)). Even though cases (2) and (5) show more accurate velocity profiles, velocity and pressure contours exhibit oscillations near the walls (top row of Figs. 14 and 15). The problems of oscillating contours can be partly resolved by increasing grid number as shown in the middle row of Figs. 14 and 15. Contours of u and v show no oscillations, while pressure contours remain unsmooth. However, by comparing the middle and bottom rows in Figs. 14 and 15, IMLE method shows even more smoother contours both for u , v , and p under the same number of grids/particles. For the comparison with the conventional particle method, IMLE method shows accurate results with smaller number of grids and particles ($nc^2 = 80^2$ and 200^2), while SPH method reported in [55] used much larger number of particles ($N = 200^2$ and 400^2). Therefore, IMLE method is said to be more accurate than the previously proposed MLE method. The numerical data of velocities $u(y)$ and $v(x)$ at the cross sections $x = 0.5$ and $y = 0.5$ are shown in Table 6.

On the other hand, computational times for cases (3), (6) and IMLE are shown in Tables 7 and 8. There are some major issues regarding to the save of computational times in each step. For calculating intermediate velocities \mathbf{u}^* , MLE method requires an additional work to accomplish the interpolation (step 5 in algorithm 1). Therefore, it is less efficient than IMLE method. In MLS interpolation, because the neighboring particles are explicitly known in IMLE method, no additional work is needed to proceed particle searching procedure in IMLE method.

For solving the MLS matrix equation, only backward elimination procedure is taken in MLS interpolation in IMLE method. In addition, since Lagrangian particles are re-meshed to Eulerian grids, there is no need to use linked-list to manage particle distribution. Therefore, IMLE method is more efficient than the previously proposed MLE method.

The most important observation to be pointed out is that both conventional particle and MLE and IMLE methods solve transient Navier–Stokes equations because the total derivative terms are solved. It is not easy for MLE method to reach steady state solutions, while application of IMLE method can get nominally steady state solutions. In other words, by computing $L2_{\mathbf{u}} = \text{RMS}(\mathbf{u}^n - \mathbf{u}^{n-1})$ or $L2_p = \text{RMS}(p^n - p^{n-1})$, where the superscripts n and $n - 1$ are time levels, IMLE method shows nominally steady state solution for lid-driven cavity flow problem at both low and high Reynolds numbers as shown in Fig. 16, while MLE method does not. To the authors' best knowledge, it is the first time for a Lagrangian-based method to show the convergence profiles. On the other hand, while the computed solutions are reaching steady state, iteration number for the application of CG to solve the PPE is decreased to one. A considerable amount of computing time can be saved. Figure 17 shows the results mentioned above. Therefore, IMLE method is clearly shown to be more stable than the previously proposed MLE method.

After illustrating the advantages of the proposed IMLE method in terms of the accuracy, efficiency and stability, the effects of numerical schemes on the accuracies of the

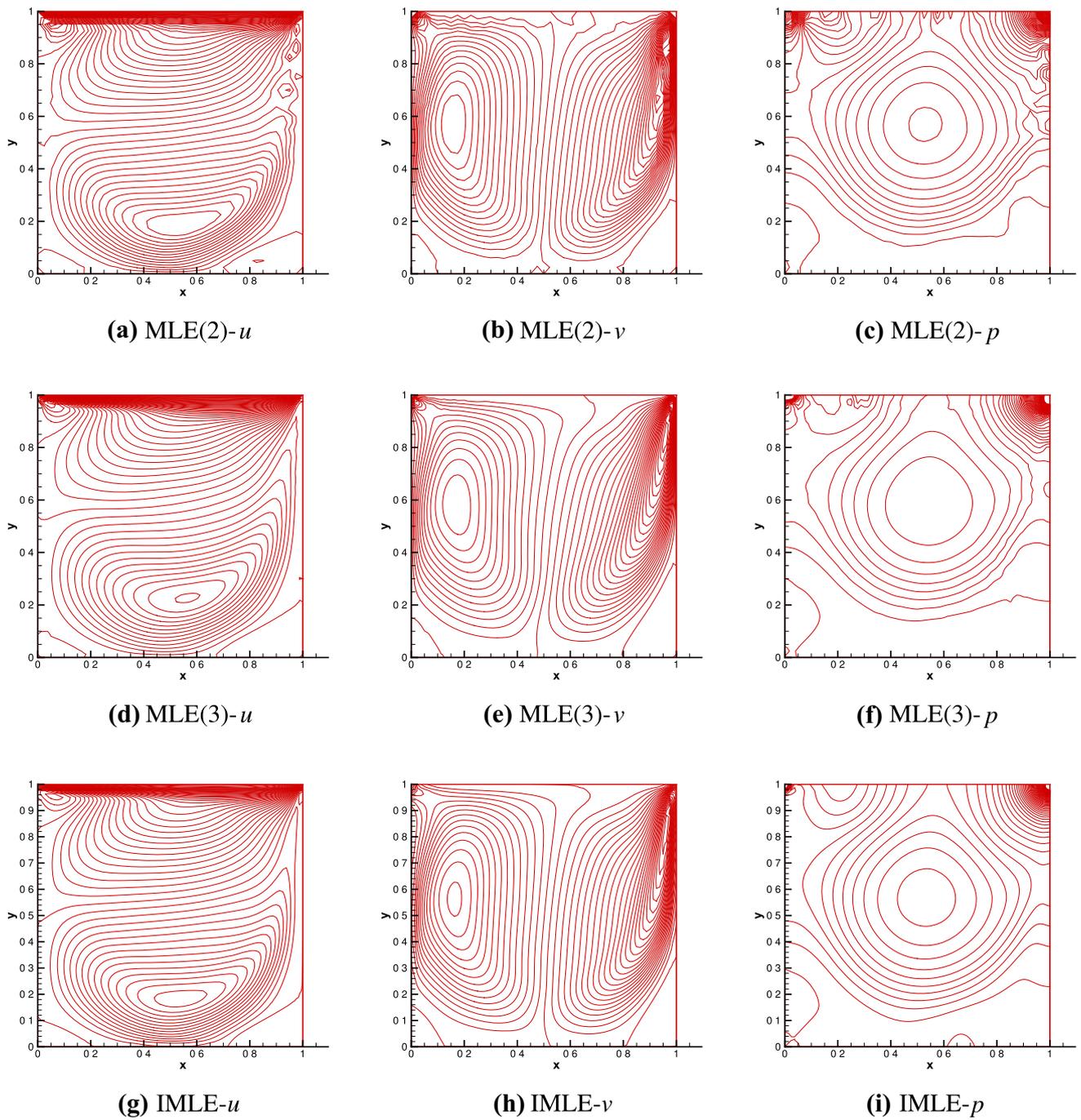


Fig. 14 Numerical results of the lid-driven cavity flow problem at $Re = 1000$ (left column: contour u ; middle column: contour v ; right column: contour p ; top row: case of MLE(2); middle row: case of MLE(3); bottom row: case of IMLE)

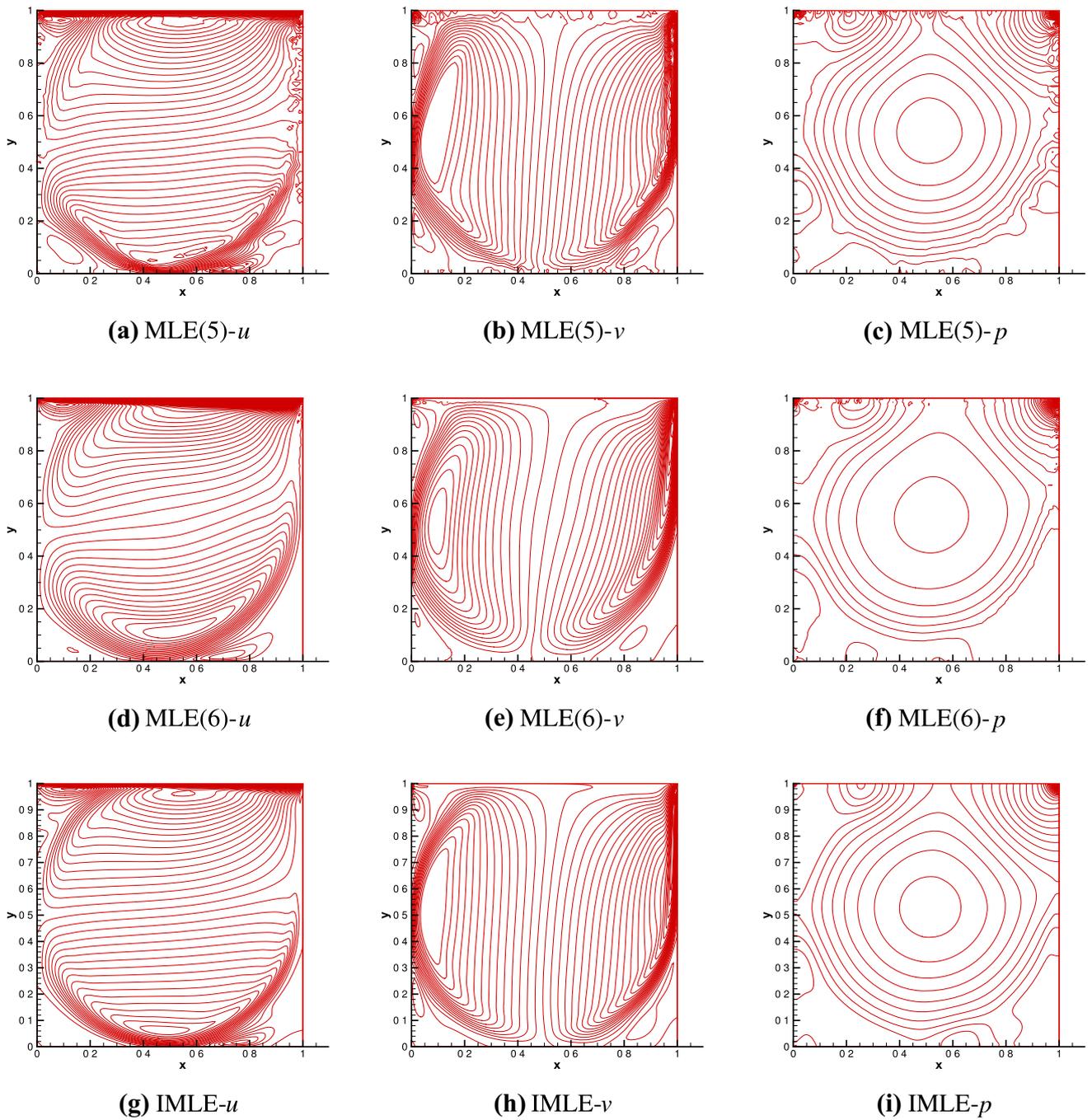


Fig. 15 Numerical results of the lid-driven cavity flow problem at $Re = 10,000$ (left column: contour u ; middle column: contour v ; right column: contour p ; top row: case of MLE(5); middle row: case of MLE(6); bottom row: case of IMLE)

Table 6 Velocities u and v at the cross sections $x = 0.5$ and $y = 0.5$ for different Reynolds numbers of the lid-driven cavity flow problem

y or x	$Re = 1000$		$Re = 10,000$	
	$u(x = 0.5, y)$	$v(x, y = 0.5)$	$u(x = 0.5, y)$	$v(x, y = 0.5)$
0.00	0.0000	0.0000	0.0000	0.0000
0.05	-0.1558	0.2257	-0.4094	0.4012
0.10	-0.2706	0.3120	-0.3627	0.3833
0.15	-0.3483	0.3485	-0.3139	0.3243
0.20	-0.3565	0.3399	-0.2676	0.2730
0.25	-0.3090	0.2956	-0.2239	0.2248
0.30	-0.2493	0.2378	-0.1818	0.1791
0.35	-0.1976	0.1812	-0.1411	0.1352
0.40	-0.1513	0.1285	-0.1012	0.0927
0.45	-0.1058	0.0777	-0.0619	0.0510
0.50	-0.0597	0.0272	-0.0226	0.0097
0.55	-0.0129	-0.0237	0.0169	-0.0317
0.60	0.0349	-0.0754	0.0576	-0.0738
0.65	0.0846	-0.1285	0.0999	-0.1174
0.70	0.1374	-0.1827	0.1446	-0.1630
0.75	0.1945	-0.2373	0.1929	-0.2113
0.80	0.2553	-0.2976	0.2458	-0.2631
0.85	0.3138	-0.3901	0.3049	-0.3191
0.90	0.3579	-0.4918	0.3724	-0.3808
0.95	0.4350	-0.3511	0.4445	-0.4620
1.00	1.0000	0.0000	1.0000	0.0000

Table 7 Computational times for the MLE(3) and IMLE at $Re = 1000$ of the lid-driven cavity flow problem

Program	Case (3)		IMLE		
	Time (s)	Prop. (%)	Time (s)	Prop. (%)	Speedup
Main	1.1796E+4	100.00	2.4201E+3	100.00	4.87
\mathbf{u}^*	2.7623E+2	2.34	1.3733E+2	5.67	2.01
MLS intrpl.	6.8301E+3	57.90	8.8174E+2	36.43	7.75
p^{n+1}	4.0819E+3	34.60	1.2853E+3	53.11	3.18

Table 8 Computational times for the MLE(6) and IMLE at $Re = 10,000$ of the lid-driven cavity flow problem

Program	Case (6)		IMLE		
	Time (s)	Prop. (%)	Time (s)	Prop. (%)	Speedup
Main	7.2239E+5	100.00	3.2185E+5	100.00	2.24
\mathbf{u}^*	1.1835E+4	1.64	9.8702E+3	3.07	1.20
MLS intrpl.	2.7382E+5	37.90	3.3744E+4	10.48	8.11
p^{n+1}	4.1168E+5	56.99	2.7375E+5	85.06	1.50

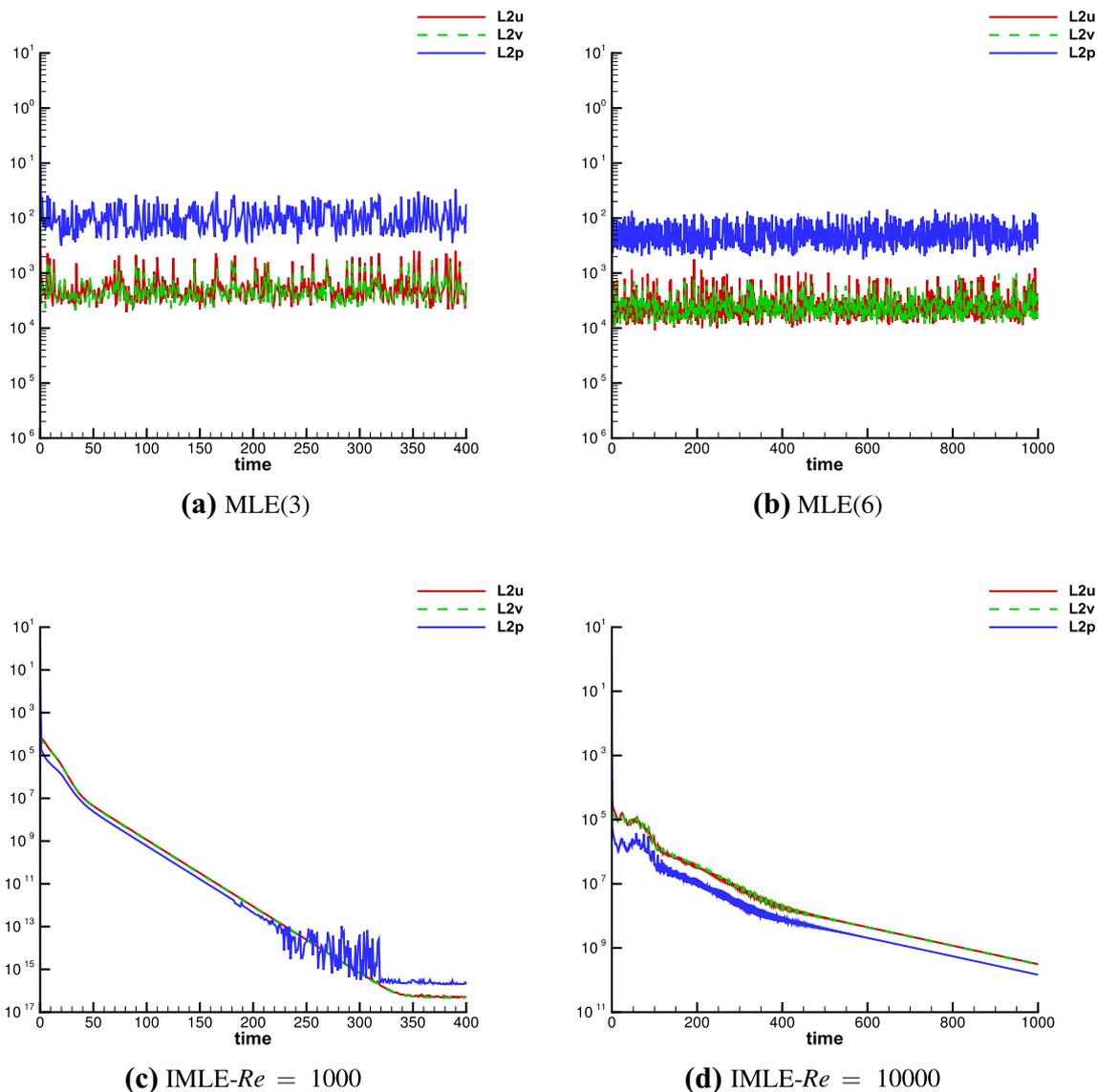


Fig. 16 Convergence profiles of the MLE (top row) and IMLE (bottom row) methods of the lid-driven cavity flow problem

velocity Laplacian (physical diffusion) term and the interpolation procedure based on the MLS method is examined. In the calculation of velocity Laplacian terms, the conventional second-order centered finite difference (FD2), the fourth-order compact finite difference (CD4), and the proposed sixth-order cell-centered combined compact finite difference (CC-CCD6) schemes are adopted. When performing MLS interpolation, the first- and the second-order accurate

schemes can be similarly derived as the derivation of the third-order scheme given in Eq. (19) by setting the truncation errors to be $O(x, y)$ and $O(x^2, y^2)$, respectively. Figure 18 compares the numerical results obtained from different Laplacian and MLS interpolation schemes. Note the notation Lap_n and MLS_m signify that the velocity Laplacian term and the MLS interpolation scheme are calculated by $2n$ -th- and m -th-order accurate schemes, where n and m are

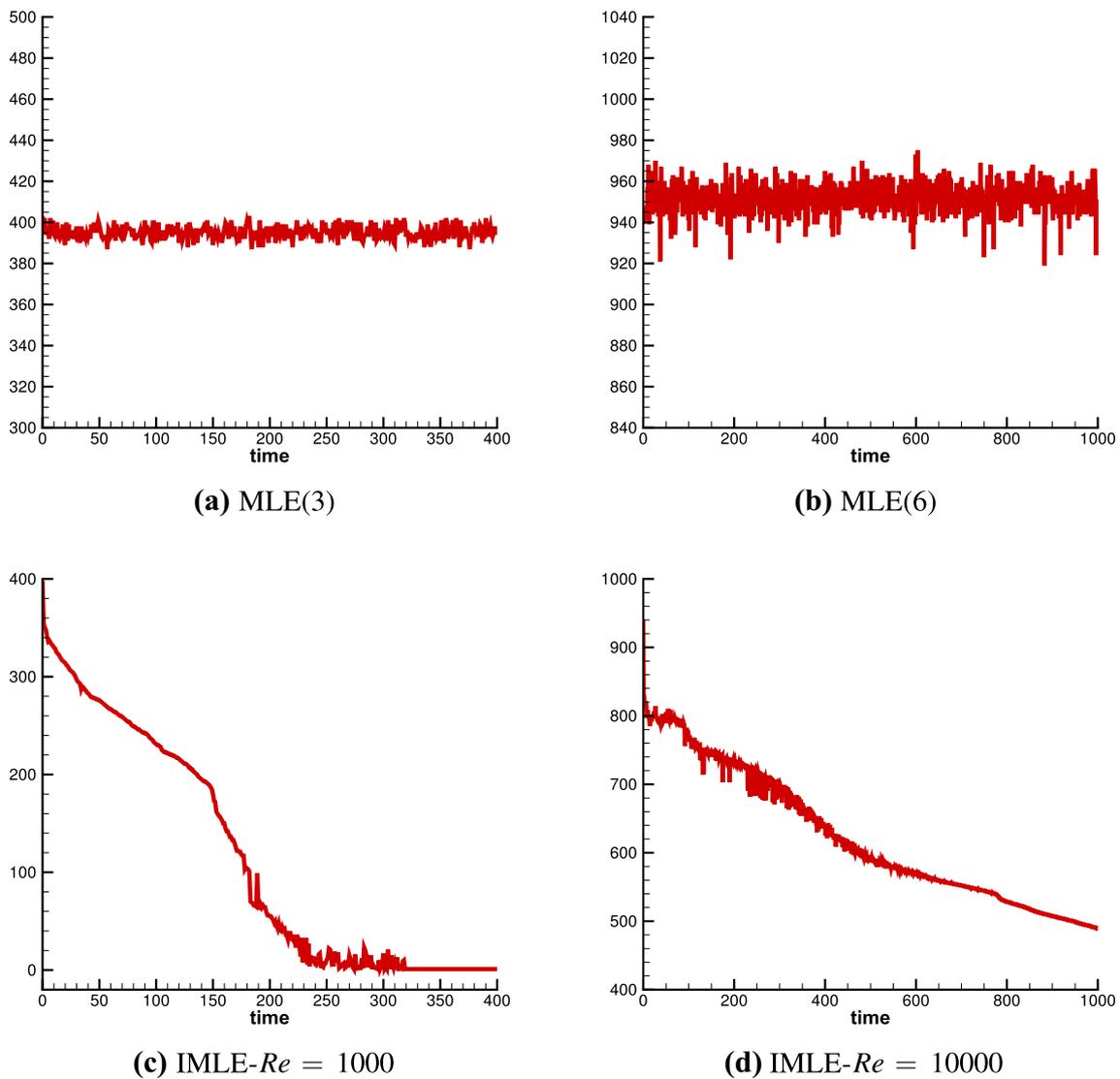


Fig. 17 Number of CG iterations of the MLE (top row) and IMLE (bottom row) methods of the lid-driven cavity flow problem

1, 2 or 3. For the same velocity Laplacian schemes (left column in Fig. 18), a slight improvement is seen when using the second-order accurate MLS interpolation scheme, as compared with the solutions obtained from the first-order scheme. In addition, there is a significant improvement in accuracy when the third-order accurate MLS interpolation scheme is adopted. On the other hand, the order of accuracy of Lapla-

cian scheme has no significant effect on the numerical results when the same MLS interpolation scheme is adopted. Seemingly, the accuracy of the numerical solution is more sensitive to the MLS interpolation scheme.

Finally, the proposed IMLE method is compared with the collocated finite volume solutions obtained from OpenFoam [76]. Here, the flow case with high Reynolds number

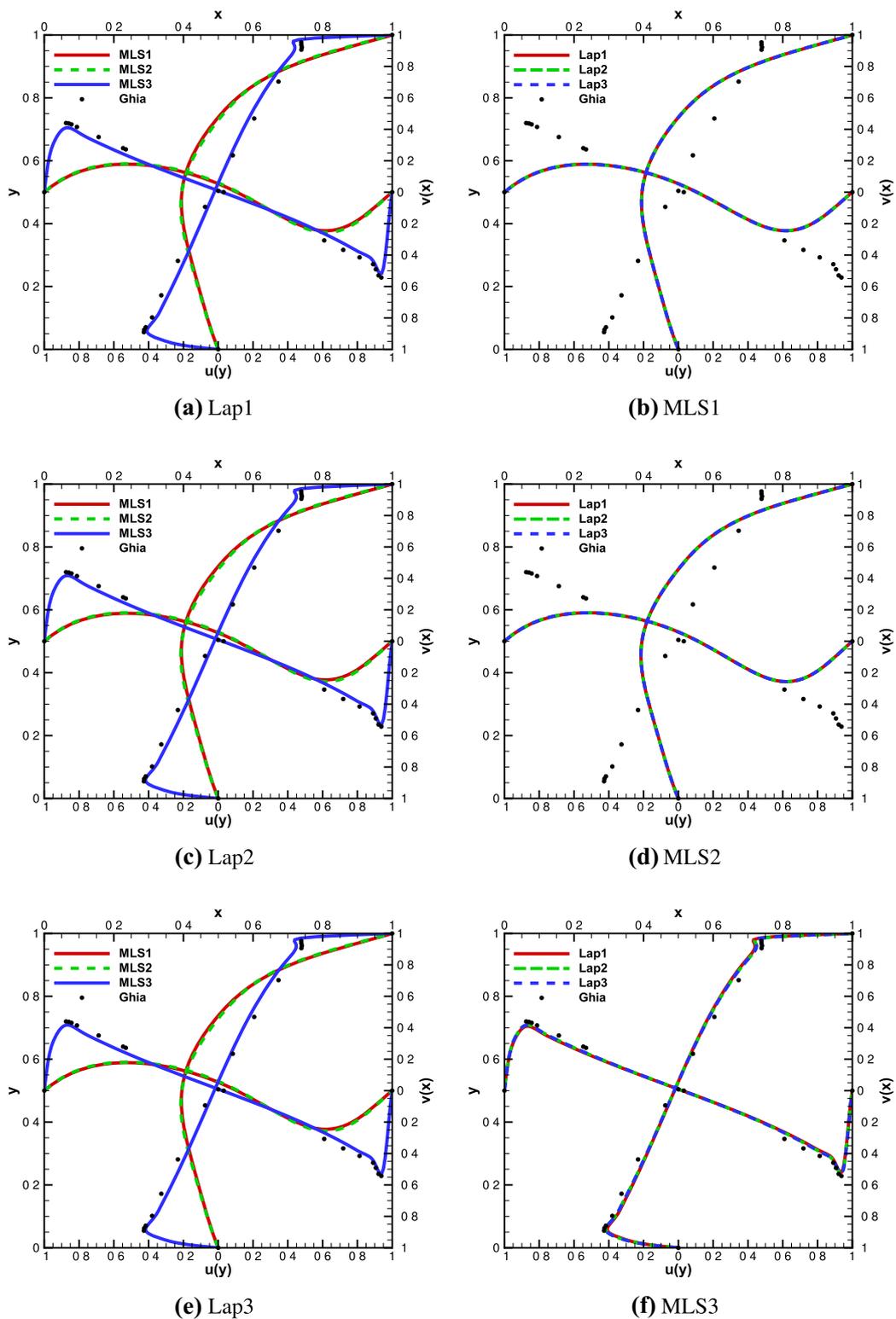
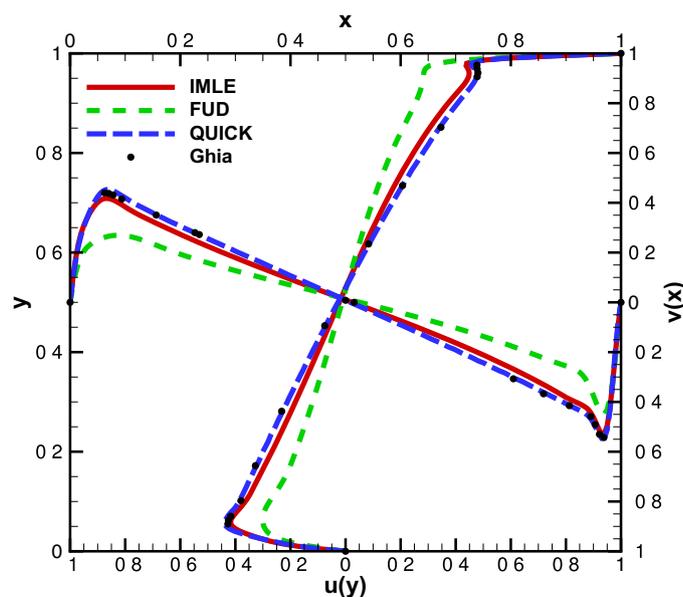


Fig. 18 Numerical results of adopting different schemes in calculating velocity Laplacian terms and performing MLS interpolation for the lid-driven cavity flow problem investigated at $Re = 10,000$. Lap1, Lap2, and Lap3 indicate that the velocity Laplacian term is approximated by

FD2, CD4, and CC-CCD6 scheme, respectively. MLS1, MLS2, and MLS3 indicate that the MLS interpolation scheme is first-, second-, and third-order accurate, respectively

Fig. 19 Velocity profiles obtained from the proposed IMLE method and the OpenFoam *icoFoam* module with different convective schemes



($Re = 10,000$) is considered. The module *icoFoam* in OpenFoam is used, whereby it solves the unsteady incompressible laminar Navier–Stokes equations using the PISO scheme. In this study, cell number $nc^2 = 200^2$, time step $\Delta t = 0.1\Delta x$ and simulation time $T = 1000$ are chosen. For OpenFoam, the numerical scheme used for discretizing the partial derivative term ($\partial \mathbf{u} / \partial t$) is Euler implicit scheme. For spatial discretization, the pressure gradient term (∇p) and the velocity Laplacian term ($\nabla^2 \mathbf{u}$) are discretized using the second-order central difference scheme, while the convection term ($\nabla \cdot \mathbf{u}\mathbf{u}$) is discretized using either the first-order upwind (FUD) or the QUICK schemes. In order to generate the structured Cartesian mesh, the *blockMesh* module is executed. Figure 19 shows that the predicted velocity profiles using the proposed IMLE method come closer to those of the third-order QUICK scheme, while the FUD scheme (from OpenFoam) shows significant numerical diffusion. Meanwhile, Fig. 20 shows that the contours of velocity components u and v predicted from the proposed IMLE method are significantly smoother than those obtained by using the QUICK scheme. Figure 21 shows the streamlines and the vorticity contours obtained from the proposed IMLE method and the OpenFoam with the QUICK convective scheme. It is obviously seen that use of the proposed IMLE method can get non-oscillating numerical solutions. The unphysical oscillations are mainly arisen due to erroneously introduced phase

and amplitude errors (or false diffusion error). However, the proposed IMLE method is more physically reasonable since moving particles are purely advected along the streamlines at their respective velocities.

5 Concluding remarks

In this study, an improved mixed Lagrangian–Eulerian (IMLE) method is successfully developed to solve incompressible Navier–Stokes equations on the basis of the previously proposed MLE method. IMLE method combines the advantages of mesh and meshless methods to yield a higher accuracy order and avoid discretizing convection terms, respectively. The key feature is that by adopting a re-meshing procedure, intermediate velocities on Eulerian grids can be calculated without introducing convection terms but with advecting Lagrangian particles and adopting MLS interpolation. In this way, high-order accurate finite difference scheme is adopted to calculate diffusion terms on Eulerian grids, while Lagrangian methods usually show low accuracy order. In order to utilize MLS interpolation for both interior, boundary, and corner cells, some special treatments for calculating intermediate velocities on boundary face particles are proposed. Weakly non-uniform and 3^2 -particle compact

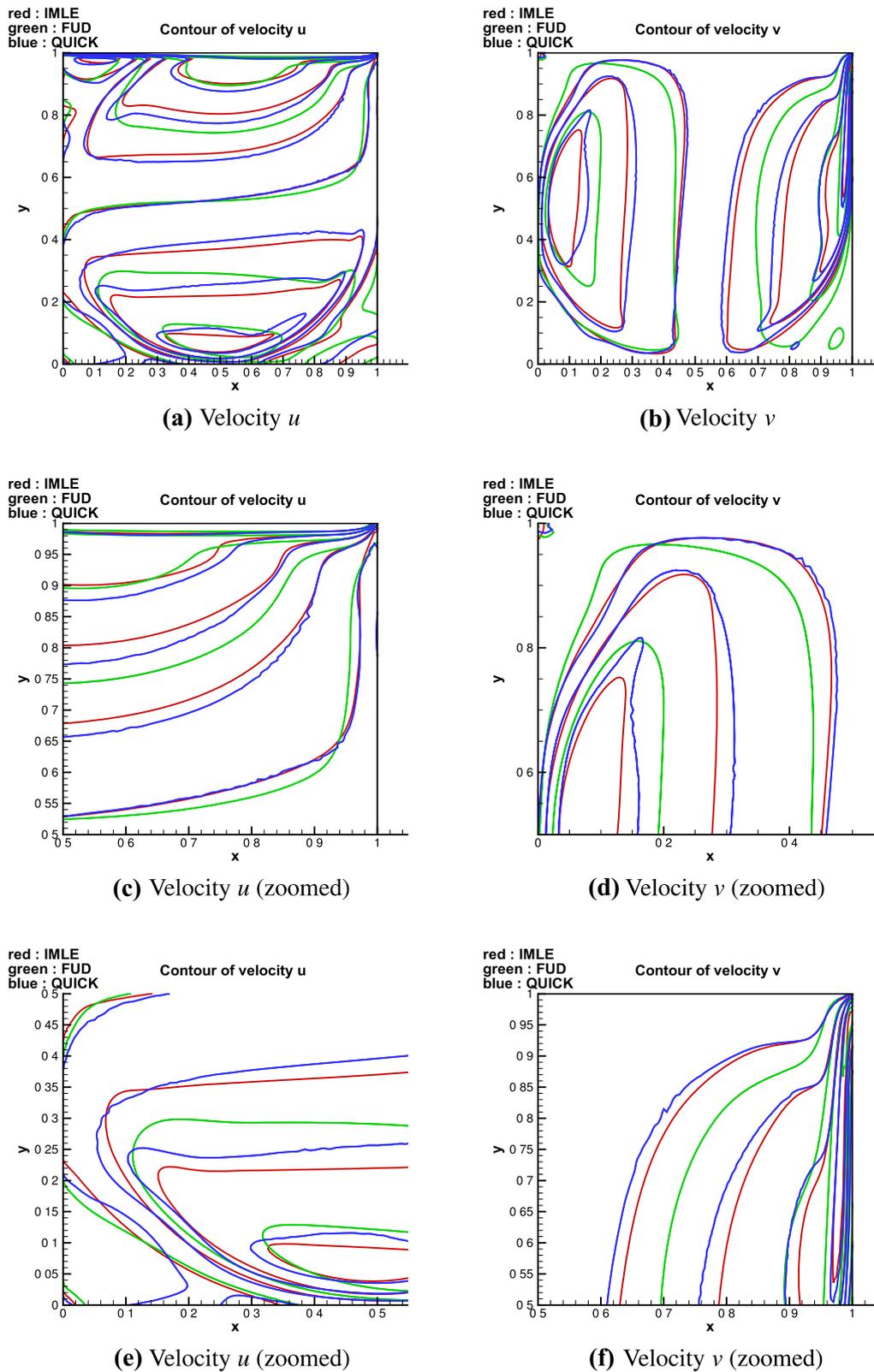


Fig. 20 Contours of the velocity components u and v obtained from the proposed IMLE method and the OpenFoam *icoFoam* module with FUD and QUICK convective schemes

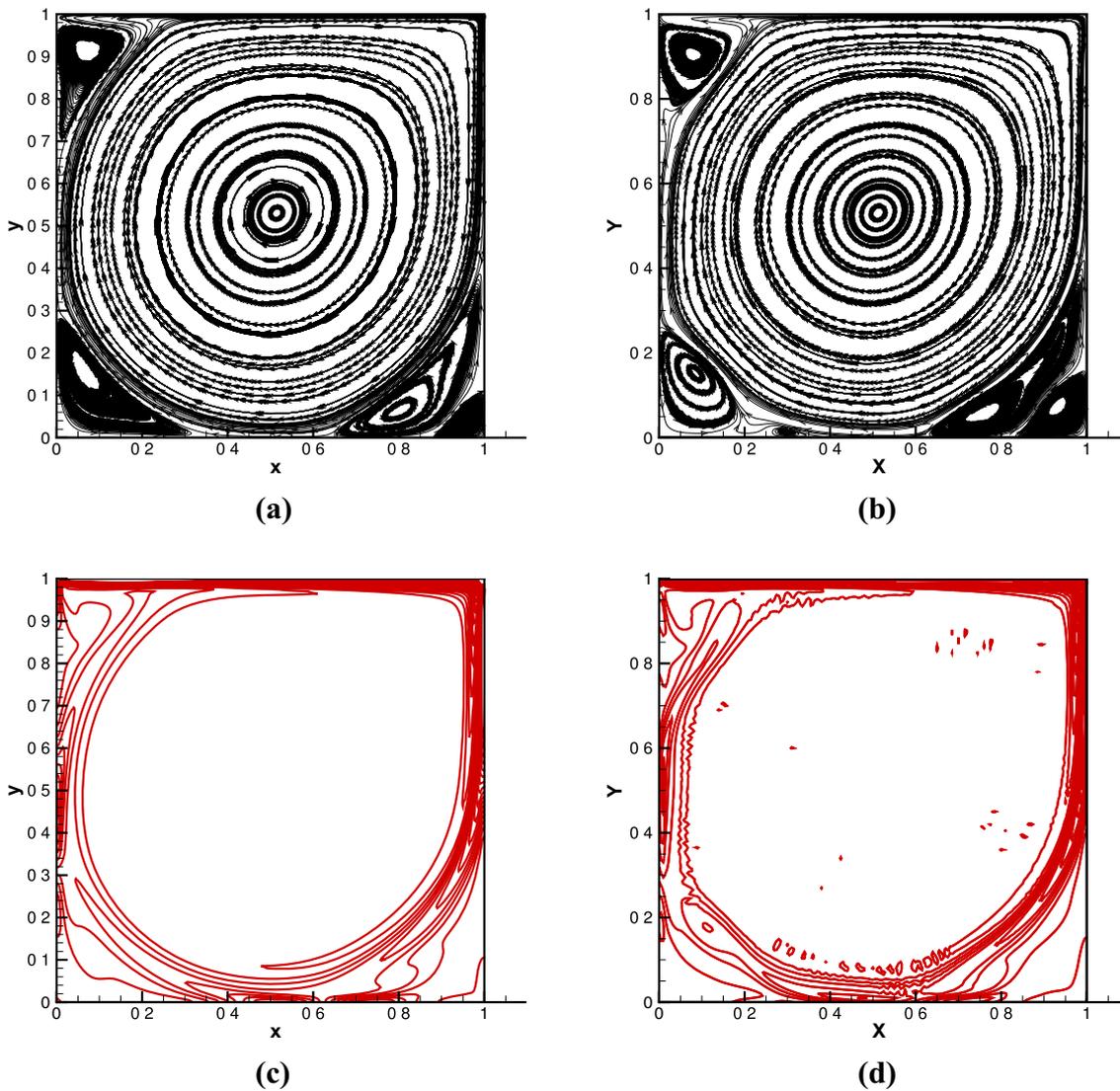


Fig. 21 Streamlines and vorticity contours obtained from the solutions of the IMLE method and the OpenFoam with the QUICK scheme. **a** Streamlines obtained from the computed IMLE solutions; **b** streamlines obtained from the OpenFoam solutions using QUICK scheme; **c** vor-

ticity contours obtained from the computed IMLE solutions; **d** vorticity contours obtained from the OpenFoam solutions using QUICK scheme

framework can improve MLS interpolation accuracy. In the last section, IMLE method shows good numerical results for different kinds of problems including periodic, inflow-outflow, and bounded flows. Finally, it is reiterated that for the fully periodic problem, our proposed IMLE method can render fourth- and second-order accuracies for velocity and pressure, respectively. Also, IMLE outperforms MLE in terms of the numerical accuracy, computational efficiency, and scheme stability. It is also worth noting that the proposed method can be easily extended to solve the three-dimensional

problems and it is under our current development. Since the proposed IMLE method adopts Eulerian grids to calculate the velocity Laplacian term and solve the PPE, one has to put in extra efforts to confront with difficulties of complex geometries inside the flow domain, irregular physical domain, and distorted free surface. For these issues, the coupled immersed boundary (IB) [77] and IMLE method, and the coupled level-set [10,29] and IMLE method are under our current development.

Funding This work was supported by the Ministry of Science and Technology (MOST) of Republic of China (R.O.C.) under the Grants MOST 106-2221-E-002-107-MY2.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest.

References

- Koshizuka S, Oka Y (1996) Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl Sci Eng* 123:421–434
- Gingold RA, Monaghan JJ (1977) Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon Not R Astron Soc* 181:375–389
- Lucy LB (1977) A numerical approach to the testing of the fission hypothesis. *Astron J* 82:1013–1024
- Hwang YH (2011) A moving particle method with embedded pressure mesh (MPPM) for incompressible flow calculations. *Numer Heat Transf B Fundam* 60:370–398
- Shibata K, Masaie I, Kondo M, Murotani K, Koshizuka S (2015) Improved pressure calculation for the moving particle semi-implicit method. *Comput Part Mech* 2:91–108
- Xu T, Kin YC (2016) Improvements for accuracy and stability in a weakly-compressible particle method. *Comput Fluids* 137:1–14
- Sanchez-Mondragon J (2016) On the stabilization of unphysical pressure oscillations in MPS method simulations. *Int J Numer Methods Fluids* 82:471–492
- Tanaka M, Cardoso R, Bahai H (2018) Multi-resolution MPS method. *J Comput Phys* 359:106–136
- Park S, Jeun G (2011) Coupling of rigid body dynamics and moving particle semi-implicit method for simulating isothermal multi-phase fluid interactions. *Comput Methods Appl Mech Eng* 200:130–140
- Ng KC, Hwang YH, Sheu TWH, Yu CH (2015) Moving particle level-set (MPLS) method for incompressible multiphase flow computation. *Comput Phys Commun* 196:317–334
- Wang L, Jiang Q, Zhang C (2017) Improvement of moving particle semi-implicit method for simulation of progressive water waves. *Int J Numer Methods Fluids* 85:69–89
- Morris JP, Fox PJ, Zhu Y (1997) Modeling low Reynolds number incompressible flows using SPH. *J Comput Phys* 136:214–226
- Cummins S, Rudman M (1999) An SPH projection method. *J Comput Phys* 152:584–607
- Shao S, Lo EYM (2003) Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Adv Water Resour* 26:787–800
- Liu GR, Liu MB (2003) Smoothed particle hydrodynamics: a mesh-free particle method. World Scientific Publishing Co. Pte. Ltd., Singapore
- Chiron L, Oger G, de Lefle M, Le Touze D (2018) Analysis and improvements of adaptive particle refinement (APR) through CPU time, accuracy and robustness considerations. *J Comput Phys* 354:552–575
- Sun PN, Colagrossi A, Marrone S, Antuono M, Zhang AM (2018) Multi-resolution delta-plus-SPH with tensile instability control: towards high Reynolds number flows. *Comput Phys Commun* 224:63–80
- Yang X, Liu M, Peng S (2014) Smoothed particle hydrodynamics and element bending group modeling of flexible fibers interacting with viscous fluids. *Phys Rev E* 90:063011
- Tamai T, Koshizuka S (2014) Least squares moving particle semi-implicit method. *Comput Part Mech* 1:277–305
- Fangyuan H, Matsunaga T, Tamai T, Koshizuka S (2017) An ALE particle method using upwind interpolation. *Comput Fluids* 145:21–36
- Zhang S, Morita K, Fukuda K, Shirakawa N (2006) An improved MPS method for numerical simulations of convective heat transfer problems. *Int J Numer Methods Fluids* 51:31–47
- Khayyer A, Gotoh H (2009) Modified moving particle semi-implicit methods for the prediction of 2D wave impact pressure. *Coast Eng* 56:419–440
- Lee BH, Park JC, Kim MH, Hwang SC (2011) Step-by-step improvement of MPS method in simulating violent free-surface motions and impact-loads. *Comput Methods Appl Mech Eng* 200:1113–1125
- Ng KC, Sheu TWH, Hwang YH (2016) Unstructured moving particle pressure mesh (UMPPM) method for incompressible isothermal and non-isothermal flow computation. *Comput Methods Appl Mech Eng* 305:703–738
- Wei H, Pan W, Rakhsha M, Tian Q, Haiyan H, Negrut D (2017) A consistent multi-resolution smoothed particle hydrodynamics method. *Comput Methods Appl Mech Eng* 324:279–299
- Obeidat A, Bordas SPA (2018) Three-dimensional remeshed smoothed particle hydrodynamics for the simulation of isotropic turbulence. *Int J Numer Methods Fluids* 86:1–19
- Qingsong T, Liu S (2017) An updated Lagrangian particle hydrodynamics (ULPH) for Newtonian fluids. *J Comput Phys* 348:493–513
- Lian YP, Zhang X, Zhou X, Ma S, Zhao YL (2011) Numerical simulation of explosively driven metal by material point method. *Int J Impact Eng* 38:238–246
- Zhang F, Zhang X, Sze KY, Lian Y, Liu Y (2017) Incompressible material point method for free surface flow. *J Comput Phys* 330:92–110
- Chen JK, Beraun JE, Carney TC (1999) A corrective smoothed particle method for boundary value problems in heat conduction. *Int J Numer Methods Eng* 46:231–252
- Shao JR, Li HQ, Liu GR, Liu MB (2012) An improved SPH method for modeling liquid sloshing dynamics. *Comput Struct* 100–101:18–26
- Zhang ZL, Liu MB (2017) Smoothed particle hydrodynamics with kernel gradient correction for modeling high velocity impact in two- and three-dimensional spaces. *Eng Anal Bound Elem* 83:141–157
- Liu MB, Li SM (2016) On the modeling of viscous incompressible flows with smoothed particle hydrodynamics. *J Hydrodyn* 28:731–745
- Zhang ZL, Liu MB (2018) A decoupled finite particle method for modeling incompressible flows with free surfaces. *Appl Math Model* 60:606–633
- Chiu PH, Sheu Tony WH (2009) On the development of a dispersion-relation-preserving dual-compact upwind scheme for convection–diffusion equation. *J Comput Phys* 228:3640–3655
- Tam CKW, Webb JC (1993) Dispersion-relation-preserving finite difference schemes for computational acoustics. *J Comput Phys* 107:262–281
- Bhumkar Y, Sheu TWH, Sengupta TK (2014) A dispersion relation preserving optimized upwind compact difference scheme for high accuracy flow simulations. *J Comput Phys* 278:378–399
- Brambley EJ (2016) Optimized finite-difference (DRP) schemes perform poorly for decaying or growing oscillations. *J Comput Phys* 324:258–274

39. Raithby GD (1976) Skew upstream differencing schemes for problems involving fluid flow. *Comput Methods Appl Mech Eng* 9:153–164
40. Patel MK, Markatos NC, Cross M (1985) Method of reducing false-diffusion errors in convection–diffusion equations. *Appl Math Model* 9:302–306
41. Carey C, Scanlon TJ, Fraser SM (1993) SUCCA—an alternative scheme to reduce the effects of multidimensional false diffusion. *Appl Math Model* 17:263–270
42. Bailey RT (2017) Managing false diffusion during second-order upwind simulations of liquid micromixing. *Int J Numer Methods Fluids* 83:940–959
43. Liu KS, Sheu Tony WH, Hwang YH, Ng KC (2017) High-order particle method for solving incompressible Navier–Stokes equations within a mixed Lagrangian–Eulerian framework. *Comput Methods Appl Mech Eng* 325:77–101
44. Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. *Math Comput* 37:141–158
45. Cueto-Felgueroso L, Colominas I, Nogueira X, Navarrina F, Casteleiro M (2007) Finite volume solvers and moving least-squares approximations for the compressible Navier–Stokes equations on unstructured grids. *Comput Methods Appl Mech Eng* 196:4712–4736
46. Chassaing J-C, Khelladi S, Nogueira X (2013) Accuracy assessment of a high-order moving least squares finite volume method for compressible flows. *Comput Fluids* 71:41–53
47. Avesani D, Dumbser M, Bellin A (2014) A new class of moving-least-squares WENO-SPH schemes. *J Comput Phys* 270:279–299
48. Ramirez L, Nogueira X, Khelladi S, Chassaing J-C, Colominas I (2014) A new high-order finite volume method based on moving least squares for the resolution of the incompressible Navier–Stokes equations on unstructured grids. *Comput Methods Appl Mech Eng* 278:883–901
49. Khayyer A, Gotoh H (2008) Development of CMPS method for accurate water-surface tracking in beaking waves. *Coast Eng J* 50:179–207
50. Tanaka M, Masunaga T (2010) Stabilization and smoothing of pressure in MPS method by quasi-compressibility. *J Comput Phys* 229:4279–4290
51. Khayyer A, Gotoh H (2011) Enhancement of stability and accuracy of the moving particle semi-implicit method. *J Comput Phys* 230:3093–3118
52. Kondo M, Koshizuka S (2011) Improvement of stability in improving particle semi-implicit method. *Int J Numer Methods Fluids* 65:638–654
53. Monaghan JJ (1994) Simulating free surface flows with SPH. *J Comput Phys* 110:399–406
54. Lee ES, Moulinec C, Xu R, Violeau D, Laurence D, Stansby P (2008) Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free method. *J Comput Phys* 227:8417–8436
55. Adami S, Hu XY, Adams NA (2013) A transport-velocity formulation for smoothed particle hydrodynamics. *J Comput Phys* 241:292–307
56. Hwang YH (2012) Assessment of diffusion operators in a novel moving particle method. *Numer Heat Transf B Fundam* 61:329–368
57. Hwang YH (2011) Smoothing difference scheme in a moving particle method. *Numer Heat Transf B Fundam* 60:203–234
58. Chu PC, Fan CW (1998) A three-point combined compact difference scheme. *J Comput Phys* 140:370–399
59. Strikwerda JC (1997) High-order-accurate schemes for incompressible viscous flow. *Int J Numer Methods Fluids* 24:715–734
60. Zhuang Y, Sun XH (2001) A high-order fast direct solver for singular poisson equations. *J Comput Phys* 171:79–94
61. Chorin AJ (1968) Numerical solution of the Navier–Stokes equations. *Math Comput* 22:745–762
62. Huang CL, Sheu TWH, Ishikawa T, Yamaguchi T (2011) Development of a particle interaction kernel for convection–diffusion scalar transport equation. *Numer Heat Transf B Fundam* 60:96–115
63. Joldes GR, Chowdhury HA, Wittek A, Doyle B, Miller K (2015) Modified moving least squares with polynomial bases for scattered data approximation. *Appl Math Comput* 266:893–902
64. Tyliczszak A (2014) A high-order compact difference algorithm for half-staggered grids for laminar and turbulent incompressible flows. *J Comput Phys* 276:438–467
65. Adami S, Hu XY, Adams NA (2012) A generalized wall boundary condition for smoothed particle hydrodynamics. *J Comput Phys* 231:7057–7075
66. Kunz P, Hirschler M, Huber M, Nieken U (2016) Inflow/outflow with Dirichlet boundary conditions for pressure in ISPH. *J Comput Phys* 326:171–187
67. Mitsume N, Yoshimura S, Murotani K, Yamada T (2015) Explicitly represented polygon wall boundary model for the explicit MPS method. *Comput Part Mech* 2:73–89
68. Li S, Liu WK (2002) Meshfree and particle methods and their applications. *Appl Mech Rev* 55:1–34
69. Liu MB, Liu GR, Lam KY (2003) Constructing smoothing functions in smoothed particle hydrodynamics with applications. *J Comput Appl Math* 155:263–284
70. Liu MB, Liu GR (2006) Restoring particle consistency in smoothed particle hydrodynamics. *Appl Numer Math* 56:19–36
71. Liu WK, Jun S, Zhang YF (1995) Reproducing kernel particle method. *Int J Numer Methods Fluids* 20:1081–1106
72. Ellero M, Serrano M, Espanol P (2007) Incompressible smoothed particle hydrodynamics. *J Comput Phys* 226:1731–1752
73. Quinlan NJ, Lobovsky L, Nestor RM (2014) Development of the meshless finite volume particle method with exact and efficient calculation of interparticle area. *Comput Phys Commun* 185:1554–1563
74. Armaly BF, Durst F, Pereira JCF, Schonung B (1983) Experimental and theoretical investigation of backward-facing step flow. *J Fluid Mech* 127:473–496
75. Ghia U, Ghia KN, Shin CT (1982) High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *J Comput Phys* 48:387–411
76. Weller HG, Tabor G, Jasak H, Fureby C (1998) A tensoral approach to computational continuum mechanics using object-oriented techniques. *Comput Phys* 12:620–631
77. Ng KC (2009) A collocated finite volume embedding method for simulation of flow past stationary and moving body. *Comput Fluids* 38:347–357