



Unstructured Moving Particle Pressure Mesh (UMPPM) method for incompressible isothermal and non-isothermal flow computation

K.C. Ng^{a,*}, T.W.H. Sheu^b, Y.H. Hwang^c

^a Center of Fluid Dynamics (CFD), Department of Mechanical Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia

^b Center for Advanced Studies in Theoretical Sciences (CASTS), National Taiwan University, Taipei, Taiwan

^c Department of Marine Engineering, National Kaohsiung Marine University, Kaohsiung 805, Taiwan

Received 18 September 2015; received in revised form 10 March 2016; accepted 10 March 2016

Available online 18 March 2016

Highlights

- Extension of MPPM particle-mesh method to handle complex flow domain.
- Extension of MPPM to handle non-isothermal buoyant flow.
- The viscous term in MPPM is treated by an implicit and consistent Laplacian model.
- No ghost particles are used to model no-slip wall boundary.
- Accuracy enhancement in MPPM can be achieved via particle adaptation.

Abstract

In this work, we intend to address the limitation of our earlier particle method, namely the Moving Particle Pressure Mesh (MPPM) method in handling arbitrary-shaped flow boundaries. The application of the Cartesian pressure mesh system adopted in our original MPPM method, which serves as the main key in recovering the divergence-free velocity condition for incompressible flow in the framework of particle method, is rather limited to rectangular flow domain. Here, the hybrid unstructured pressure mesh is adopted to remove the geometrical constraint of our earlier MPPM method. Coupled with the moving particle strategy in the Moving Particle Semi-implicit (MPS) method, the new method is named as the Unstructured Moving Particle Pressure Mesh (UMPPM) method in the current work. A consistent Laplacian model, namely the Consistent Particle Method (CPM) recently reported in the open literature is incorporated as well in the framework of UMPPM for discretizing the viscous term on the scattered particle cloud, while its implicit form is solved in the current work for overall robustness. Finally, we shall verify our UMPPM method with a series of benchmark solutions (for isothermal and non-isothermal flows) available from the literatures, including those obtained from the commercial code. It is appealing to find that the numerical solutions of UMPPM compare well with the benchmark solutions. In some cases, the accuracy of our UMPPM is better than that of the existing particle method such as Smoothed Particle Hydrodynamics (SPH).

© 2016 Elsevier B.V. All rights reserved.

Keywords: Moving Particle Semi-implicit (MPS); Consistent Particle Method (CPM); Moving Particle Pressure Mesh (MPPM); Particle-In-Cell (PIC); Unstructured mesh; Incompressible flow

* Corresponding author. Tel.: +60 389212020x6478.

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

1. Introduction

Since the proposal of the particle method, namely the Moving Particle Semi-implicit (MPS) method by Koshizuka and Oka [1], this numerical technique has been gaining significant attentions amongst the CFD practitioners since the past decade. In MPS, which is rather similar to the Smoothed Particle Hydrodynamics (SPH) approach, the Navier–Stokes Equation (NSE) is solved in the Lagrangian mesh-free manner with a set of interacting particles. One of the important advantages of particle method is the exact treatment of convection term in the flow transport equation by advecting the fluid particle in the Lagrangian manner. To date, MPS has been successfully applied to a wide range of engineering applications, including turbomachinery problem [2,3], flow boiling problem [4], injection molding problem [5], solidification problem [6], liquid dripping problem [7] and mixing problem [8,9] to name a few.

Most of the engineering flow domains are geometrically complex. This has translated into several numerical challenges in MPS, particularly on developing a robust technique to avoid the deficiency of particle number density of wall particles (denoted as $n_{i,wall}$) taking part in the solution process of the Pressure Poisson Equation (PPE). The ghost particle approach such as that of placing multiple layers (number of layer is dependent on the value of radius of influence) of ghost particles near the wall particle is the most common approach in MPS. While it is trivial to accomplish this in a simple geometry (say a rectangular flow domain), the flexibility of ghost particle approach is questionable particularly when the geometrically complex flow domain is encountered. There are three approaches available in the literature to address this problem. In the first approach, the particles are placed in their voxel positions (or the stair-case approach). Although this technique could assure uniform initial particle number density (n_0) throughout the flow field, the modeled boundary is not continuous and it may differ from the actual geometry of the flow boundary (unless a very fine resolution is used at the boundary). The related MPS works utilizing this approach to treat non-rectangular flow boundary can be found in [10–14,8,9]. The second approach is similar to the body-fitted mesh generation approach in the mesh-based method, whereby the wall particles are placed along the flow boundary. The ghost particles are then displaced accordingly to adhere to the geometry of the flow boundary. This approach can be traced back from the earlier MPS works by Koshizuka et al. [15], Koshizuka and Oka [16], Kouh [17] and Shakibaeinia and Jin [18], whereby a slanted plane has been modeled. Application of this approach on more complex flow boundary can be found from the works by Nakanishi et al. [2] in modeling the geometry of the Pelton bucket and Sun et al. [19,3] in modeling the geometry of a cylindrical stator. Although this approach can be used to address the limitation of the first approach (i.e. geometric conformance), discussion on how to ensure a uniform distribution of initial particle number density (n_0), which is a key requirement for incompressible flow problem, is unclear. The more recent (or the third) approach works by negating the use of ghost particles. Here, a smooth and continuous function (dependent only on the distance from the wall) is used to replace the calculation of particle number density contributed by the wall. Fine-tuning of this function, however, is required if one wishes to extend this method to handle problems with arbitrarily shaped boundaries. The related works can be found in [20] and the very recent MPS work reported by Li et al. [21].

Besides that, it is challenging to enforce the no-slip boundary condition at the wall if the shape of the flow boundary is complex. The earlier proposal of Koshizuka et al. [15] suggested the modeling of no-slip wall by placing multiple layers of zero-velocity ghost particles adjacent to the wall particles. Owing to the inaccuracy of this conventional approach, this has prompted [22] in using the mirror image particle to enforce the no-slip condition. A similar work can be seen from the works by Zhang et al. [23] and Lee et al. [24]. Owing to the fact that the locations of mirror image particles are dynamically updated in each time step, an increase of computational load can be expected. Also, it is very challenging while placing the mirror image particles in the vicinity of sharp-angled boundaries. Interested readers may refer to the recent work of Akimoto [25] for a more detailed explanation on the implementation of this method.

In spite of the fact that different ways have been proposed in MPS to handle flows involving complex geometry, its robustness in solving practical flow problem involving arbitrarily complex geometry is still worth for further investigation. Few years ago, we have been working on a hybrid Lagrangian–Eulerian method, namely the Moving Particle Pressure Mesh (MPPM) approach [26,27] by combining the benefits of both Eulerian and particle methods in simulating incompressible flow. The main idea is to treat the pressure as the Eulerian variable, which is in contrast with the original proposal in MPS method whereby the pressure and velocity are stored on the Lagrangian particles. In MPPM method, the mass conservation equation (hence PPE) is solved on the Eulerian background mesh and hence a

divergence-free velocity condition can be assured on the mesh level. Owing to the fact that information such as particle number density is not required in the entire solution process of MPPM, this has surmounted several computational deficiencies in the existing MPS method. However, due to the nature of the Eulerian mesh used in MPPM which is essentially Cartesian, this has hampered its applicability in practical flow problems. In order to address the no-slip boundary at the wall, the wall particles are simply represented by a series of stationary particles placed on the wall boundary without the use of mirror image (or ghost) particle. This approach is preferable, as discussed in our recent MPPM method for multiphase flow (see the Moving Particle Level-Set (MPLS) method by Ng et al. [28]), owing to the fact that it is computationally cheaper and simpler to implement. However, an accurate Laplacian model must be sought in order to address the deficiency of number of particles near the wall. Still, in most of our computational work conducted earlier by using MPPM [26,27], a numerically inconsistent Laplacian model (e.g. the MPS Laplacian model of Koshizuka et al. [15], which is $O(h^{-1})$ in irregular particle layout where h is the particle spacing) has been used to discretize the viscous term. Although the development of a new Laplacian model has been reported in our recent MPLS work [28] in order to reduce the leading error of the MPS Laplacian model, its order of accuracy is only $O(h^0)$.

There are several papers available in the open literature addressing the use of hybrid Eulerian–Lagrangian approach in fluid flow simulation. In fact, the MPPM shares a very similar idea with the Particle-In-Cell (PIC) method [29,30], whereby the advection of *massless* particles is adopted to treat the convective term. According to the very recent PIC work reported by Kelly et al. [30], the excessive numerical diffusion inherited in the original advection procedure of the PIC method can be circumvented via incrementing the particle velocity based on the change in the grid velocity. This procedure, however, requires one to perform numerous numerical interpolations between the grid and the particle levels (see Section 3.4 in [30]). A very recent paper reported by Matsunaga et al. [31] utilizes a hybrid approach as well, whereby the Lagrangian advection procedure is used only to solve the species concentration equation to minimize the numerical diffusion, while the flow equations are solved on the Eulerian mesh via the finite volume method. Due to the truncation error inherited from the time discretization scheme implemented in the particle advection procedure, particle distribution error will occur and the highly anisotropic particle spacing may cause numerical instability. In the recent works of Kelly et al. [30] and Matsunaga et al. [31], certain tunable particle regularization/redistribution procedures must be employed to *artificially* shift the positions of particles to improve spacing uniformity. Although these particle regularization procedures can be omitted by recovering the mass property on each Lagrangian particle such as that reported by Liu et al. [32], PPE must be expressed on both mesh and particle levels and this will ultimately increase the computational cost. As one shall witness later, in contrast with PIC and other hybrid procedures reported above, MPPM solves the momentum equations directly on the Lagrangian particles (similar to the fully Lagrangian methods such as SPH and MPS), without performing numerical interpolation from the mesh level as practiced in the above literatures. Numerical interpolation takes place, however, from the particle to mesh level, in order to obtain the mesh face velocities that appeared as the source term of PPE. Due to the employment of massless particles in MPPM, we have similarly noticed the anisotropic spacing of fluid particles as simulation progresses. This problem is resolved by performing the particle addition/deletion procedure (discussed later), without artificially shifting the positions of the particles.

In the current work, we aim to extend the MPPM solver to handle arbitrary-shaped flow boundaries. The Cartesian mesh originally embedded in MPPM can be used to handle a geometrically complex flow domain with modifications, for instance, by the Immersed Boundary (IB) method. However, it involves complex data interpolation particularly at the flow boundary (see [33,34]). Owing to this, the body-fitted unstructured mesh is chosen as the candidate to replace the Cartesian pressure mesh embedded in our original MPPM method. In the current work, the associated modifications in our original MPPM solver which are implemented to accommodate the more robust unstructured pressure mesh will be explained in detail. Also, we recognize the limitation of our Laplacian model adopted in our previous work [26–28], which is numerically inconsistent. Therefore, we shall investigate on a consistent Laplacian model (i.e. the Consistent Particle Method (CPM)) recently proposed by Koh et al. [35], whereby consistency is ensured via the Generalized Finite Difference (GFD) and the Least Square (LS) methods. In order to enhance its robustness, an implicit form of CPM is considered in the computation of viscous stress term reported in current work, which is never reported before in open literature. Also, we shall further extend our method to solve non-isothermal flows. Finally, we shall verify our new Unstructured Moving Particle Pressure Mesh (UMPPM) method with the benchmark solutions available for Taylor–Green decaying vortex problem, lid-driven flow problems in several cavities, inflow–outflow problems and some natural convection problems.

2. Governing equation

The two-dimensional incompressible flow is considered in the current work, whereby the fluid flow is governed by the mass conservation (continuity) equation:

$$\nabla \bullet \vec{u} = 0 \quad (1)$$

and the momentum conservation equation:

$$\rho \frac{D\vec{u}}{Dt} = -\nabla P + \mu \nabla^2(\vec{u}) + \vec{S}. \quad (2)$$

Applications involving heat transfer can be analyzed by solving the energy equation alongside with Eqs. (1) and (2):

$$\rho C_p \frac{DT}{Dt} = k \nabla^2(T). \quad (3)$$

In the above equations, ρ is the fluid density, P is the fluid pressure, μ is the fluid dynamic viscosity, \vec{S} is the momentum source vector, $\vec{u} = \langle u, v \rangle$ is the fluid velocity vector, T is the fluid temperature, k is the fluid thermal conductivity and C_p is the fluid specific heat. In order to account for buoyancy force, the Boussinesq approximation is used:

$$\vec{S} = -\rho \beta \vec{g}(T - T_o), \quad (4)$$

where β is the fluid thermal expansion coefficient, \vec{g} is the gravitational acceleration vector and T_o is the reference temperature.

3. Numerical method

3.1. Moving Particle Semi-implicit (MPS) and Moving Particle Pressure Mesh (MPPM) methods

In MPS, the flow domain is filled (or discretized) with finite number of particles i . By integrating the momentum equation (Eq. (2)), the velocity of particle i at time level $n + 1$ can be determined by:

$$\vec{u}_i^{n+1} = \vec{u}_i^n + \frac{1}{\rho_i} \int \left(\mu \nabla^2(\vec{u})_i + \vec{S}_i - \nabla P_i \right) dt. \quad (5)$$

By using the conventional Moving Particle Semi-implicit (MPS) method [1], the diffusion/source terms and pressure gradient term are commonly discretized by using the first-order explicit and implicit schemes, respectively:

$$\vec{u}_i^{n+1} = \vec{u}_i^n + \frac{\Delta t}{\rho_i} \left(\mu_i \nabla^2(\vec{u})_i^n + \vec{S}_i^n - \nabla P_i^{n+1} \right), \quad (6)$$

and the new position of the particle i can be updated as:

$$\vec{r}_i^{n+1} = \vec{r}_i^n + \Delta t \vec{u}_i^{n+1}. \quad (7)$$

The pressure field P^{n+1} which appeared in Eq. (6) must be recovered in such a way that it satisfies the continuity equation (1). As such, the fractional-step approach is commonly adopted. For the first step, the intermediate velocity of particle i is computed via:

$$\vec{u}_i^* = \vec{u}_i^n + \frac{\Delta t}{\rho_i} \left(\mu_i \nabla^2(\vec{u})_i^n + \vec{S}_i^n \right). \quad (8)$$

In order to take the flow convection effect into account, the particle i is advected in the following manner:

$$\vec{r}_i^* = \vec{r}_i^n + \Delta t \vec{u}_i^*. \quad (9)$$

Here, the superscript * denotes the intermediate time level. In the second step, the effect of pressure gradient is recovered by:

$$\vec{u}_i^{n+1} = \vec{u}_i^* - \frac{\Delta t}{\rho_i} \nabla P_i^{n+1}. \quad (10)$$

The Pressure Poisson Equation (PPE) can then be derived by simply putting Eq. (10) into the divergence form and applying the continuity equation ($\nabla \bullet \vec{u}_i^{n+1} = 0$) to yield:

$$\frac{\Delta t}{\rho_i} \nabla^2 P_i^{n+1} = \nabla \bullet \vec{u}_i^*. \quad (11)$$

Eq. (11) is called the “divergence-free” condition in the MPS work reported by Tanaka and Masunaga [36]. Apart from the fact that a smooth pressure field can be attained (due to the relative smooth source term in Eq. (11)), the particle number density (denoted as n in MPS literature) is not required. However, the conservation of volume of a fluid particle (hence the particle number density, n), which is an essential requirement for incompressible fluid calculation, cannot be assured as the time progresses. This has prompted Tanaka and Masunaga [36] to modify the source term of the PPE into the following form:

$$\frac{\Delta t}{\rho_i} \nabla^2 P_i^{n+1} = \nabla \bullet \vec{u}_i^* + \gamma \frac{1}{\Delta t} \frac{n_i^o - n_i^k}{n_i^o}. \quad (12)$$

Here, γ is the tuning parameter, n_i^o and n_i^k denote the particle number density of particle i at time level 0 (i.e. initial condition) and time level k ($k > 0$), respectively. A similar idea can be found from other MPS works reported by Kondo and Koshizuka [37], Lee et al. [24], Ng and Ng [9], Ng et al. [8] and Natsui et al. [38]. While the above formulation (Eq. (12)) may lead to a smoother pressure field and volume conservation, the effort of calibrating the tuning parameter γ is cumbersome. On the other hand, a large amount of computational effort is needed to reconstruct the coefficient matrix of the PPE at each time step due to the possible repositioning of fluid particles.

In fact, if one treats the fluid particle as an observation point rather than that which is carrying a certain mass/volume (material point), one can get rid of Eq. (12) to obtain the new pressure field. Inspired by the physical nature of pressure in an incompressible flow condition (i.e. lacking of the convective effect), Hwang [26] has argued that pressure should be treated as an Eulerian variable (as proposed in his Moving Particle Pressure Mesh (MPPM) method). Therefore, the Poisson equation of pressure (Eq. (11)) can be solved on a fixed Eulerian pressure mesh (say mesh with centroid location P) to retrieve the new pressure field:

$$\frac{\Delta t}{\rho_P} \nabla^2 P_P^{n+1} = \nabla \bullet \vec{u}_P^*, \quad (13)$$

after one has calculated the intermediate velocities and positions of all fluid particles via Eqs. (8)–(9). Here, the intermediate velocities are interpolated to the mesh level (hence mesh-based velocities) in order to construct the source term of the PPE (RHS of Eq. (13)). More importantly, the coefficient matrix of the PPE expressed in Eq. (13) is fixed (due to fixed pressure mesh employed), which could lessen the overall computational effort.

It is worth to mention here that the idea of interpolating the mesh-based intermediate velocities from the advected velocities stored in Lagrangian points (or fluid particles) in MPPM is somewhat analogous to that of the Semi-Lagrangian (SL) method proposed by Stam [39]. Instead of tracking the particles' velocity and position at each time step as implemented in MPPM, the advection procedure in SL method works by simply applying the backward trajectory approach to determine the departure location of a particle that arrives at the interpolating mesh point. The intermediate velocity at the interpolating mesh point is then prescribed to that at the departure location (obtained via interpolation from intermediate velocities stored in neighboring mesh). While this method is computationally cheaper than MPPM, the truncation error is large (numerically diffusive as reported by Jin and Chen [40]) and the departure location may be located outside of the computational domain [41].

Similar to MPS, upon solving the PPE in MPPM (see Eq. (13)), the particle velocity can be corrected via Eq. (10) and its new position can be updated in accordance with Eq. (7).

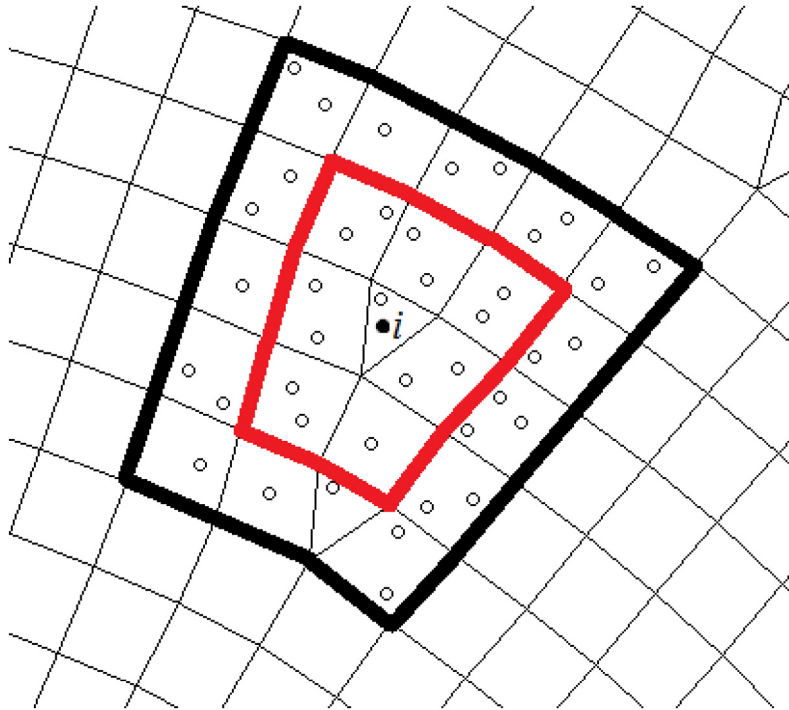


Fig. 1. Particle i (solid circle) surrounded by neighboring particle j (empty circles) in CPM. The pressure meshes contained in the thick red loop are classified as the first neighbors of the local pressure mesh containing particle i . Searching of neighboring particle is performed on pressure meshes contained in the thick black loop (which is consisting of the first and second neighbors of the local pressure mesh containing particle i). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Improvement of MPPM

3.2.1. Implicit and consistent Laplacian model

In most of the existing studies in MPS or MPPM, the diffusive term (Laplacian of velocity) is treated explicitly for the ease of computation (e.g. [1,26]). However, as the flow problem involving highly viscous fluid or small particle diameter (or higher pressure mesh resolution in MPPM), this method is simply inefficient because it delimits the allowable time step size for numerical stability. Implicit algorithm is therefore considered in the current work:

$$\vec{u}_i^* = \vec{u}_i^n + \frac{\Delta t}{\rho_i} \left(\mu_i \nabla^2 (\vec{u}_i^* + \vec{S}_i^n) \right). \quad (14)$$

The Laplacian model such as that implemented in MPS [1]:

$$\nabla^2 (\vec{u}_i^*) = \frac{2d}{\sum_{j \neq i} w(|\vec{r}_j - \vec{r}_i|) |\vec{r}_j - \vec{r}_i|^2} \sum_{j \neq i} w(|\vec{r}_j - \vec{r}_i|) (\vec{u}_j^* - \vec{u}_i^*) \quad (15)$$

can be used to discretize the diffusive term in Eq. (14), where j is the index for neighboring particle of local particle i . The above implicit form of Laplacian model has been recently considered by Fukuzawa et al. [42] in simulating highly viscous non-Newtonian flow. Here, $w(|\vec{r}_j - \vec{r}_i|)$ is taken as the standard kernel function in MPS: $w(|\vec{r}_j - \vec{r}_i|) = R/|\vec{r}_j - \vec{r}_i| - 1$, i.e. Eq. (3) reported in [15]. R is the influence radius, normally taken as 2–4 times of initial particle diameter (see [26,15]).

Notwithstanding the fact that the above Laplacian model (especially in its explicit form) has been used extensively in MPS literature, it has been mathematically shown by Ng et al. [27] that the above formulation is numerically inconsistent in irregular particle layout. A similar finding has been reported by Souto-Iglesias et al. [43] as well. Recently, Koh et al. [35] have addressed this problem by introducing a new Laplacian model (they named it as

Consistent Particle Method (CPM)), whereby consistency is ensured via the Taylor series. In the current work, the CPM model is expressed in its implicit form:

$$\nabla^2(\vec{u})_i^* = \sum_{j \neq i} w_j^2 \left[(c_1 + e_1)h_j + (c_2 + e_2)k_j + (c_3 + e_3)\frac{h_j^2}{2} + (c_4 + e_4)h_jk_j + (c_5 + e_5)\frac{k_j^2}{2} \right] (\vec{u}_j^* - \vec{u}_i^*). \tag{16}$$

Here, the neighboring particle j in CPM is searched from the neighboring pressure meshes as outlined in Fig. 1. Here, the searching operation is performed up to the second neighbor of the local pressure mesh containing particle i , which is somewhat analogous to the influence radius value recommended by Koh et al. [44], i.e. $R = 2.1l_0$ (l_0 is the initial particle diameter in MPS). In the above CPM model, the coefficients c_n, e_n ($n = 1, 2 \dots 5$) can be easily found from the Least Square (LS) method via minimizing the square of errors with the gradient components (see Eq. (20) in [35]). The outcome of the minimization is quoted below from the subsequent work by Koh et al. [44]:

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \\ c_1 & c_2 & c_3 & c_4 & c_5 \\ d_1 & d_2 & d_3 & d_4 & d_5 \\ e_1 & e_2 & e_3 & e_4 & e_5 \end{bmatrix} = \begin{bmatrix} \sum_{j \neq i} w_j^2 h_j^2 & \sum_{j \neq i} w_j^2 h_j k_j & 0.5 \sum_{j \neq i} w_j^2 h_j^3 & \sum_{j \neq i} w_j^2 h_j^2 k_j & 0.5 \sum_{j \neq i} w_j^2 h_j k_j^2 \\ \sum_{j \neq i} w_j^2 h_j k_j & \sum_{j \neq i} w_j^2 k_j^2 & 0.5 \sum_{j \neq i} w_j^2 h_j^2 k_j & \sum_{j \neq i} w_j^2 h_j k_j^2 & 0.5 \sum_{j \neq i} w_j^2 k_j^3 \\ 0.5 \sum_{j \neq i} w_j^2 h_j^3 & 0.5 \sum_{j \neq i} w_j^2 h_j^2 k_j & 0.25 \sum_{j \neq i} w_j^2 h_j^4 & 0.5 \sum_{j \neq i} w_j^2 h_j^3 k_j & 0.25 \sum_{j \neq i} w_j^2 h_j^2 k_j^2 \\ \sum_{j \neq i} w_j^2 h_j^2 k_j & \sum_{j \neq i} w_j^2 h_j k_j^2 & 0.5 \sum_{j \neq i} w_j^2 h_j^3 k_j & \sum_{j \neq i} w_j^2 h_j^2 k_j^2 & 0.5 \sum_{j \neq i} w_j^2 h_j k_j^3 \\ 0.5 \sum_{j \neq i} w_j^2 h_j k_j^2 & 0.5 \sum_{j \neq i} w_j^2 k_j^3 & 0.25 \sum_{j \neq i} w_j^2 h_j^2 k_j^2 & 0.5 \sum_{j \neq i} w_j^2 h_j k_j^3 & 0.25 \sum_{j \neq i} w_j^2 k_j^4 \end{bmatrix}^{-1}. \tag{17}$$

Here, h_j and k_j are $(x_j - x_i)$ and $(y_j - y_i)$, respectively. w_j is the weight function in CPM (taken as $w_j = 1/|\vec{r}_j - \vec{r}_i|^3$) as recommended by Gao [45] after performing numerical testing on several kernel functions. The inverse matrix (RHS of Eq. (17)) results in other coefficients as well such as a_n, b_n and d_n ($n = 1, 2 \dots 5$) which can be used to evaluate terms involving first derivative and cross derivative. At this point, according to our knowledge, the current work is the first attempt whereby the viscous stress term is discretized by the implicit CPM method.

3.2.2. The use of unstructured pressure mesh

3.2.2.1. *Pressure Poisson Equation (PPE)*. The earlier MPPM works have witnessed their applicability in simulating single-phase [26,27] and multi-phase flow [28]. Nevertheless, the computations are limited to flow applications involving rectangular flow domain. The bottleneck is due to the nature of the pressure mesh employed in the original MPPM work which is essentially Cartesian. Obviously, one may extend the Cartesian pressure mesh system to handle flow involving complex boundary [34] by using the Immersed Boundary (IB) method. Nonetheless, this method requires complex data interpolation at the boundary which may degrade the flow accuracy in the vicinity of the flow boundary.

From our point of view, the more practical solution would be employing the body-fitted unstructured pressure mesh system on the entire flow domain. As such, the discretization procedure of the PPE in MPPM must be modified accordingly. In the current work, we intend to generalize the discretization procedure of the PPE so that it is applicable

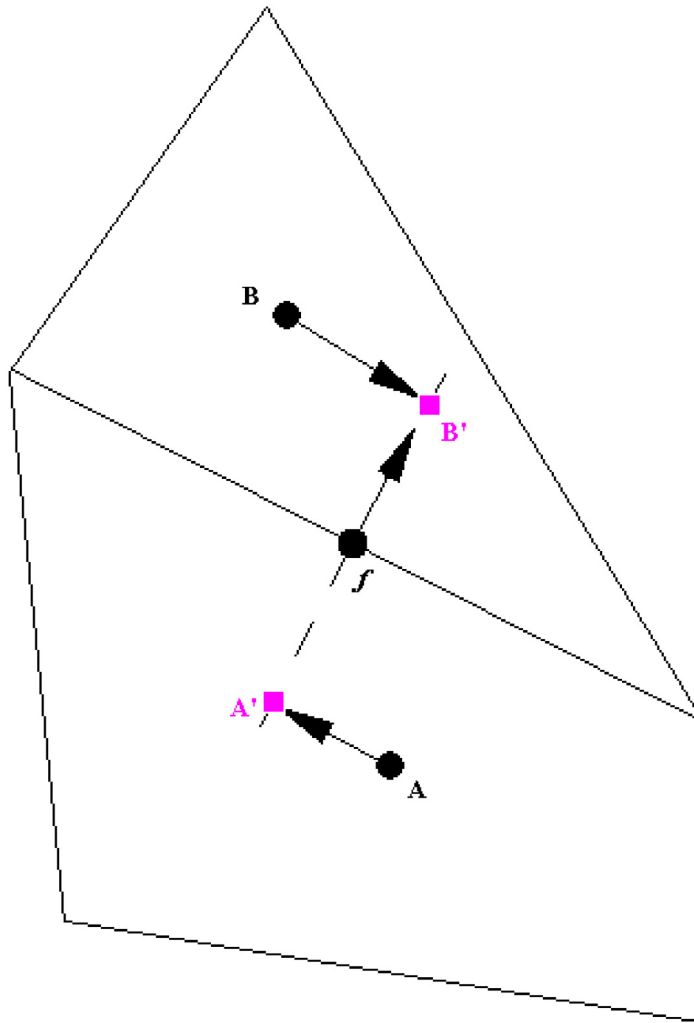


Fig. 2. Schematic diagram of two pressure meshes with centroids A and B sharing the same face f . Position vectors \vec{AA}' and \vec{BB}' are perpendicular to position vectors \vec{fA}' and \vec{fB}' , respectively.

in the hybrid unstructured pressure mesh environment. We express the PPE (Eq. (13)) in its volume integral form:

$$\int \nabla^2 P_P^{n+1} dV = \int \frac{\rho P}{\Delta t} \nabla \cdot \vec{u}_P^* dV. \tag{18}$$

Upon applying the divergence theorem, the following area integral equation is obtained:

$$\int \nabla P_f^{n+1} \cdot d\vec{A} = \frac{\rho P}{\Delta t} \int \vec{u}_f^* \cdot d\vec{A}. \tag{19}$$

The above equation can be discretized accordingly as:

$$\sum_f \nabla P_f^{n+1} \cdot \vec{A}_f = \frac{\rho P}{\Delta t} \sum_f \vec{u}_f^* \cdot \vec{A}_f \tag{20}$$

where the summation is performed on each mesh face f . Fig. 2 illustrates the configuration of two pressure meshes sharing a common face f . In this case, the area vector \vec{A}_f is pointing to the pressure mesh B .

While it is straightforward to discretize the LHS of Eq. (20) when the mesh is orthogonal (e.g. $\vec{AA}' = \vec{BB}' = \vec{0}$), unstructured mesh may give rise to the well-known issue of mesh non-orthogonality. In order to circumvent this issue,

Jasak [46] has decomposed the area vector on the LHS of Eq. (20) into its orthogonal and non-orthogonal components (denoted by subscript *o* and *no*, respectively):

$$\nabla P_f^{n+1} \bullet \vec{A}_f = \frac{P_B^{n+1} - P_A^{n+1}}{\|\vec{AB}\|} \|\vec{A}_{f,o}\| + \nabla P_f^n \bullet \vec{A}_{f,no}. \tag{21}$$

From his work, the vector $\vec{A}_{f,o}$ is parallel to \vec{AB} . Jasak [46] has studied the numerical accuracy and stability of three different methodologies in estimating the vector $\vec{A}_{f,o}$. The non-orthogonal term (second term of the RHS of Eq. (21)) is then treated as an extra source term in the PPE (on top of the mass imbalance term), which may cause numerical instability when the mesh is highly irregular.

In the current work, we adopt a more intuitive strategy (see Fig. 2) inspired by Niceno [47], whereby the LHS of Eq. (20) is discretized as:

$$\nabla P_f^{n+1} \bullet \vec{A}_f = \frac{P_{B'}^{n+1} - P_{A'}^{n+1}}{\|\vec{A'B'}\|} \|\vec{A}_f\|. \tag{22}$$

Due to the fact that pressure variables are stored at the centroids *A* and *B*, the gradient projection method (similar to those employed in [48–51]) can be used to interpolate the pressure values at the imaginary positions *A'* and *B'*:

$$P_{A'}^{n+1} = P_A^{n+1} + \nabla P_A^{n+1} \bullet \vec{AA'} \tag{23}$$

$$P_{B'}^{n+1} = P_B^{n+1} + \nabla P_B^{n+1} \bullet \vec{BB'}. \tag{24}$$

Substituting Eqs. (23), (24) into Eq. (22) gives:

$$\nabla P_f^{n+1} \bullet \vec{A}_f = \frac{P_B^{n+1} - P_A^{n+1} + \nabla P_B^{n+1} \bullet \vec{BB'} - \nabla P_A^{n+1} \bullet \vec{AA'}}{\|\vec{A'B'}\|} \|\vec{A}_f\|. \tag{25}$$

The pressure gradient at the mesh centroid can be evaluated by using the divergence theorem, or the Least-Square method by adopting the computational stencils similar to that illustrated in Fig. 3. In the current work, the latter method is preferred to get rid of the pressure interpolation at the mesh face *f* as required in the divergence theorem. Upon minimizing the square of errors with each gradient component ($\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}$), the following equation can be obtained for each mesh *P*:

$$\begin{bmatrix} \frac{\partial P^{n+1}}{\partial x_P} \\ \frac{\partial P^{n+1}}{\partial y_P} \end{bmatrix} = \begin{bmatrix} \sum_j (x_j - x_P)^2 & \sum_j (x_j - x_P)(y_j - y_P) \\ \sum_j (x_j - x_P)(y_j - y_P) & \sum_j (y_j - y_P)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_j (x_j - x_P)(P_j^{n+1} - P_P^{n+1}) \\ \sum_j (y_j - y_P)(P_j^{n+1} - P_P^{n+1}) \end{bmatrix}. \tag{26}$$

From the above equation, it is important to note that the matrix inversion is performed only once (because the pressure mesh is fixed) and the source term is constantly updated at each time step in order to evaluate the pressure gradient at mesh *P*. The pressure gradient terms ∇P_A and ∇P_B appeared in Eq. (25) can then be evaluated by using the similar method outlined in Eq. (26). Due to the involvement of neighboring mesh pressure *P_j* in the pressure gradient calculation, the coefficient matrix for the PPE will involve the second neighbor of mesh *P* (see Fig. 4), thereby enlarging the bandwidth of the coefficient matrix of the PPE as compared to that of our previous work [26–28].

As mentioned in Section 3.1, the intermediate face velocities that appeared on the RHS of the discretized PPE (Eq. (20)) are interpolated from the intermediate particle velocities in MPPM. There are many interpolation procedures available. In particle method (e.g. MPS), the most common approach is the weighted averaging technique similar to that of the Shepard interpolation procedure:

$$\vec{u}_f^* = \frac{1}{\sum_j w(|\vec{r}_j - \vec{r}_f|)} \sum_j \vec{u}_j^* w(|\vec{r}_j - \vec{r}_f|). \tag{27}$$

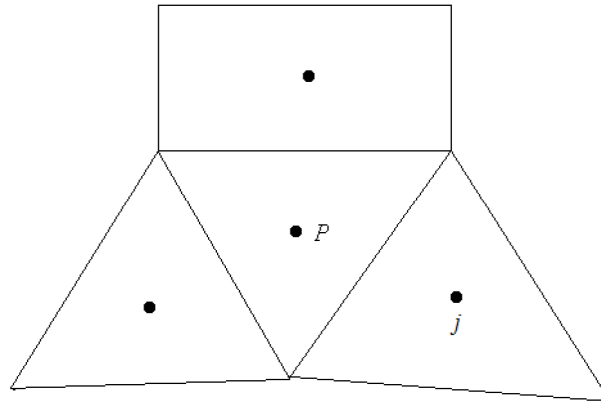


Fig. 3. Computational stencil used to evaluate the pressure gradient at mesh P with neighbors j by using the least-square method.

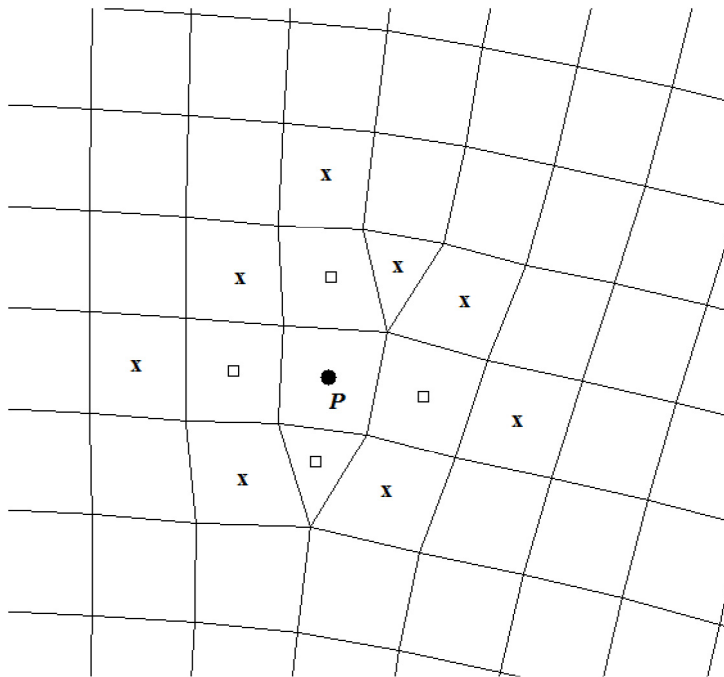


Fig. 4. Computational stencil for PPE at mesh P . \square : first neighbor; x : second neighbor.

Here, the standard kernel function in MPS such as $w(|\vec{r}_j - \vec{r}_f|) = R/|\vec{r}_j - \vec{r}_f| - 1$ [26,38] and other non-standard kernel functions such as those proposed by Yoon et al. [52] and Shakibaeinia and Jin [53] may be considered. Although the above procedure is somehow computationally cheaper than other more advanced methods such as the Moving Least Square (MLS) method [54,55], we found that the overall quality of the flow solution is very sensitive to the accuracy of the interpolated face velocity \vec{u}_f^* . Owing to this reason, the MLS technique with a quadratic basis function:

$$R = [1 \quad x \quad y \quad xy \quad x^2 \quad y^2] \tag{28}$$

is adopted in the current work. Here, we impose a local coordinate system centered at point of interest (say point f , see Fig. 5). The locally fitted intermediate velocity field in the vicinity of face f reads:

$$\vec{u}^*(x, y) = P(A^{-1}B) \tag{29}$$

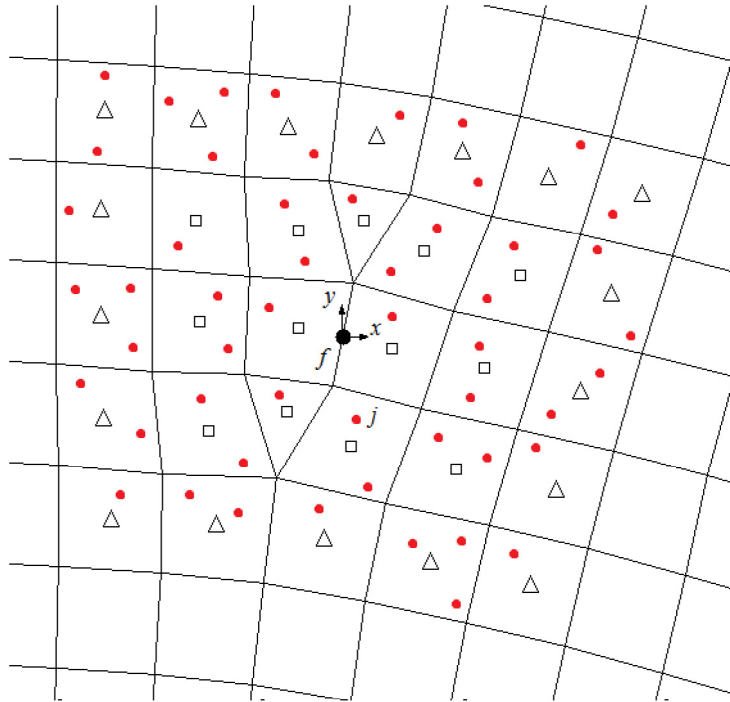


Fig. 5. Interpolation of face velocity at point f by using neighboring particle j (solid red circle). Searching of neighboring particle j is performed in the neighboring pressure meshes categorized as first neighbors (marked as empty square) and second neighbors (marked as empty upper triangle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where

$$A = \sum_j w (|\vec{r}_j - \vec{r}_f|) R^T R \tag{30}$$

$$B = \sum_j w (|\vec{r}_j - \vec{r}_f|) \vec{u}_j^* R^T. \tag{31}$$

Let $A^{-1}B = [m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6]^T$. Obviously, the interpolated face velocity \vec{u}_f^* can be obtained by simply setting $x = 0$ and $y = 0$ in the basis function appeared in Eq. (29) to yield:

$$\vec{u}_f^* = [1 \ 0 \ 0 \ 0 \ 0 \ 0] (A^{-1}B) = m_1. \tag{32}$$

Correspondingly, the partial derivatives such as $\vec{u}_{f_x}^*, \vec{u}_{f_y}^*$ can be recovered whenever necessary:

$$\vec{u}_{f_x}^* = m_2 \tag{33}$$

$$\vec{u}_{f_y}^* = m_3. \tag{34}$$

At this point, it is important to note that the neighboring particle j is searched from the neighboring pressure meshes, which are consisting of first neighbors and second neighbors as illustrated in Fig. 5. In order to account for the influence of distance between the neighboring particle j and the interpolating point (point f in this case), Tabbara et al. [54] have proposed an exponentially decaying kernel function with an adjustable parameter c to control the rate of descent. In the current work, we found that the tuning-free kernel function originally proposed by Koshizuka et al. [15] in MPS works well, with a slight modification to account for the non-uniformity of the unstructured pressure

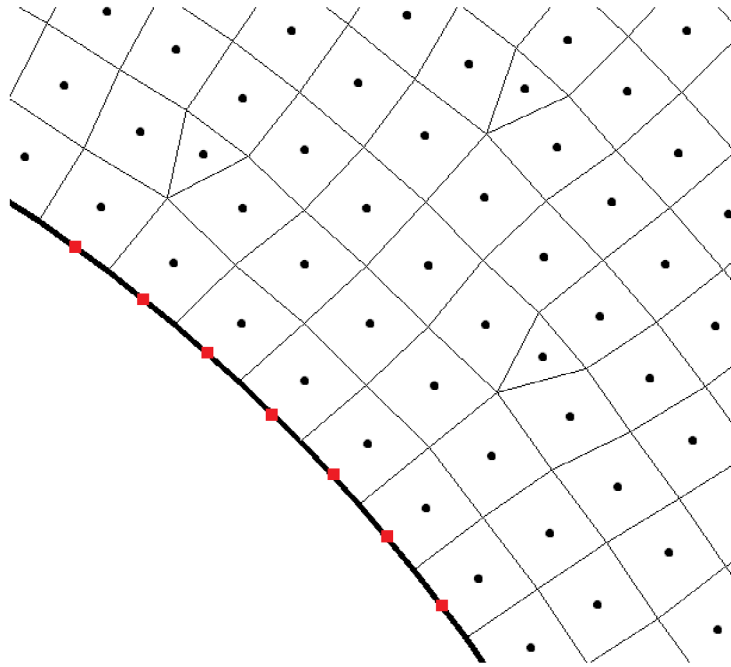


Fig. 6. Initial arrangement of fluid particle (solid black circle) and boundary particle (solid red square). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

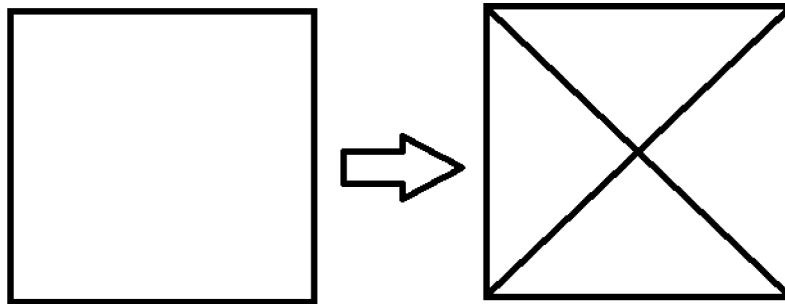


Fig. 7. Decomposition of a square element into four triangular elements for the Taylor–Green flow computation.

mesh:

$$w(|\vec{r}_j - \vec{r}_f|) = \frac{1}{\sqrt{\left(\frac{x_j - x_f}{x_m}\right)^2 + \left(\frac{y_j - y_f}{y_m}\right)^2}} - 1. \quad (35)$$

Here, $x_m = \text{Max}|x_j - x_f| + \overline{\Delta_x}$ and $y_m = \text{Max}|y_j - y_f| + \overline{\Delta_y}$. $\overline{\Delta_k}$ is the mean value of spatial lengths (in the k -direction) of the two pressure meshes straddling on face f .

Once the intermediate face velocities are obtained, the discrete PPE (Eq. (20)) is solved by the preconditioned BiCGSTAB method [56]. In order to ensure mass conservation, the PPE is solved until the L2-error norm is below 1×10^{-15} . It is important to note here that the simultaneous equations arisen from the implicit discretization of viscous term are solved by using the same sparse matrix solver implemented for PPE.

3.2.2.2. Velocity correction and particle adaptation in MPPM. Upon solving the PPE, the velocity of particle i can be corrected accordingly as outlined in Eq. (10). During the velocity correction stage, the pressure gradient of particle i can be calculated from the MLS procedure (see Eqs. (33), (34)), by making use of the new pressure values stored

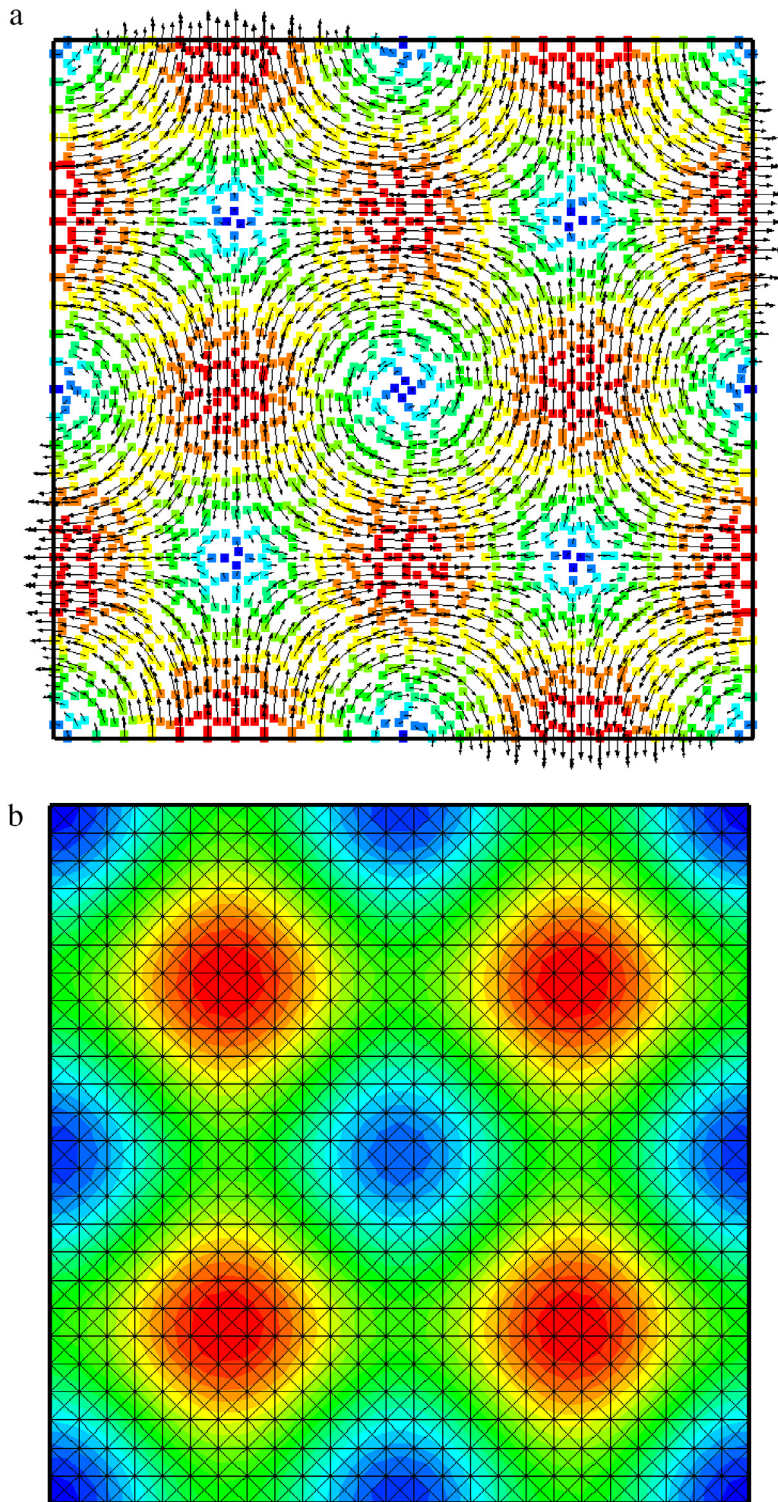


Fig. 8. (a) Particle speed $\in [0, 0.2038]$ m/s and (b) Mesh pressure $\in [-0.021, 0.021]$ Pa for the Taylor–Green flow at $t = 2$ s. $N_m = 2500$.

in the neighboring pressure meshes. In the conventional MPS method, the artificial repulsive force term (e.g. the minimum pressure model by Koshizuka et al. [15] and CMPS model by Khayyer and Gotoh [57]) is added to the

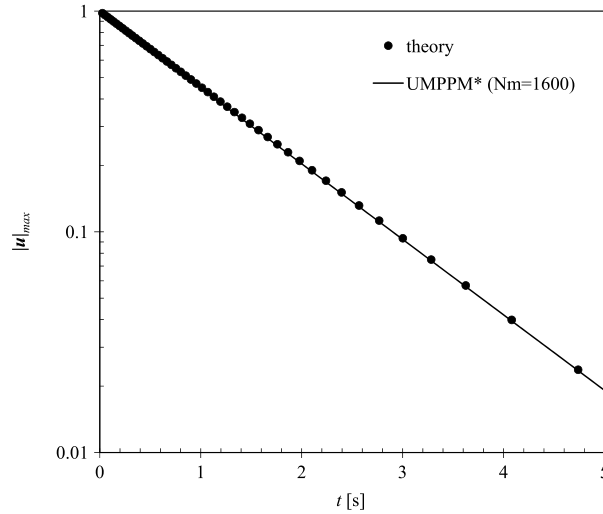


Fig. 9. Decay of maximum speed for the Taylor–Green flow. $N_m = 1600$.

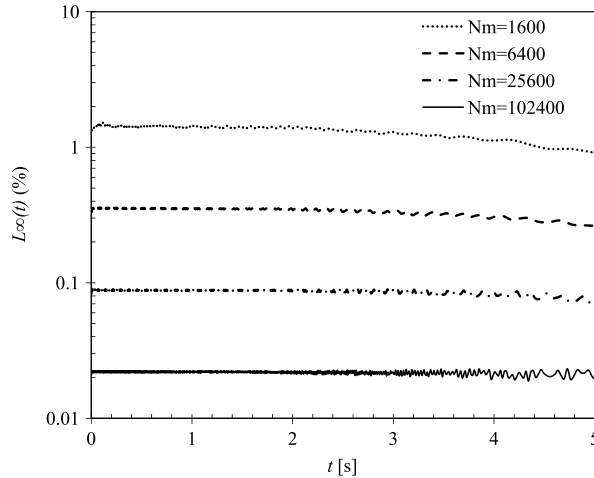


Fig. 10. Relative error of maximum speed predicted at various particle resolutions.

pressure gradient term to ensure numerical stability. The pressure gradient term in MPPM, however, is retained in its original form and no artificial treatment is required.

In the standard MPS method, collision model (such as that proposed by Lee et al. [24]; Shakibaeinia and Jin [53]; Natsui et al. [38]) is usually employed if the moving particles are too close to each other. From their work, the particle velocity \vec{u}_i^{n+1} is modified accordingly to account for the effect of collision. In MPPM method, by getting rid of the material constraint on the moving particles, particles which are close to each other ($\delta < \sqrt{\alpha}d_c$) are simply deleted as reported in the original MPPM work [26]. Here, $d_c = \sqrt{\Delta_x^2 + \Delta_y^2}$ where Δ_x and Δ_y are the spatial lengths of a pressure mesh in the x - and y -direction. α is taken as 0.09 in the original MPPM work. The particle deletion procedure is permissible because particles are merely acting as observation points (used to interpolate the intermediate face velocities in the mesh level as in Eq. (32)).

In the event of detecting an empty pressure mesh (mesh with no fluid particles), a particle is added at the mesh centroid. The velocity of the new particle i is updated accordingly as:

$$\vec{u}_i^{n+1} = \vec{u}_p^{n+1} = \vec{u}_p^* - \frac{\Delta t}{\rho_P} \nabla P_P^{n+1}. \tag{36}$$

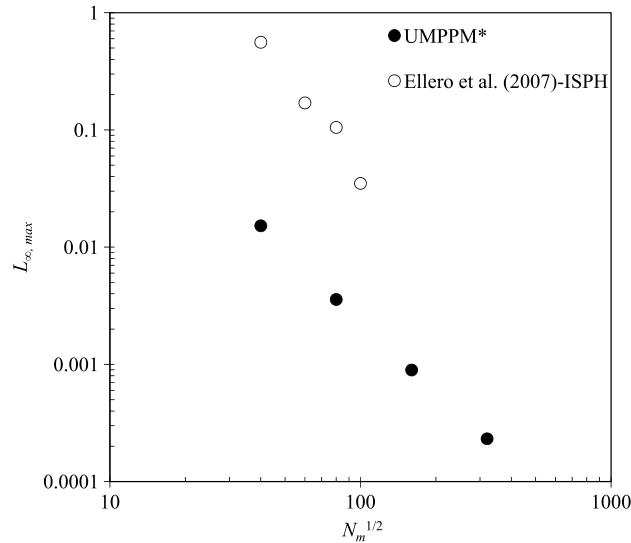


Fig. 11. Convergence of $L_{\infty,max}$ for UMPPM* and ISPH [58]. $L_{\infty,max}$ is the maximum value of $L_{\infty}(t)$ within $0 < t < 5$ s.

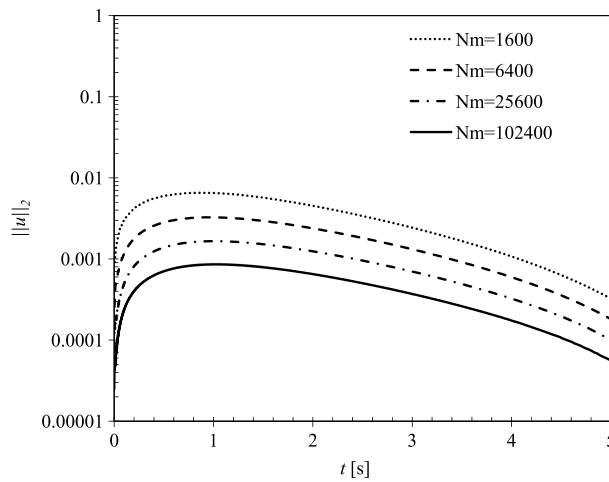


Fig. 12. The L_2 -norm of x -velocity field predicted at various particle resolutions.

Here, \vec{u}_P^* is the intermediate velocity at mesh centroid P , which is interpolated from the neighboring intermediate particle velocity \vec{u}_j^* by using the MLS procedure similar to that used to approximate the intermediate face velocity discussed in Eq. (32). The term ∇P_P^{n+1} can be evaluated via the least-square method outlined in Eq. (26).

3.3. Solution procedure of UMPPM

In general, the solution procedure of the new UMPPM method can be summarized as:

- Step 1: Build the unstructured pressure mesh on the domain of interest.
- Step 2: Insert the fluid particle at the mesh centroid and boundary particle at the centroid of the boundary face (see Fig. 6). The positions of the boundary particle remain unchanged during the simulation.
- Step 3: Evaluate the inverse matrix (first term of the RHS of Eq. (26)) for each pressure mesh. The coefficients that appeared in the inverse matrix are needed for pressure gradient calculation.
- Step 4: Construct the coefficient matrix of PPE (see Eqs. (20), (25) and (26)).
- Step 5: Calculate the time step size (Δt) according to the prescribed CFL value.

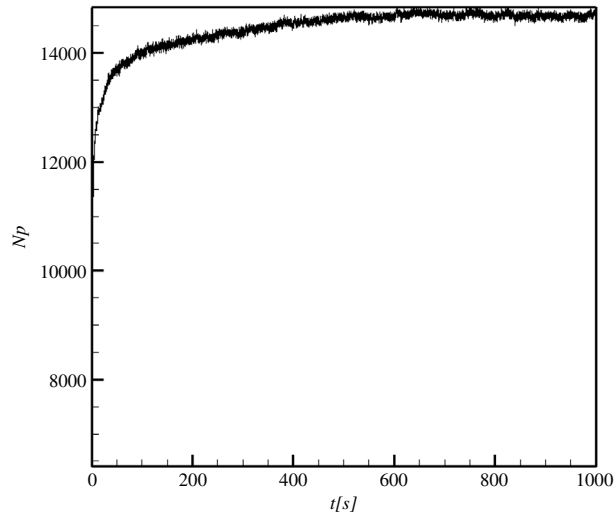


Fig. 13. The changes of total number of particles N_p for the lid-driven flow calculation. Case: UMPPM ($\alpha = 0.09$) with 80×80 pressure mesh. $Re = 10\,000$.

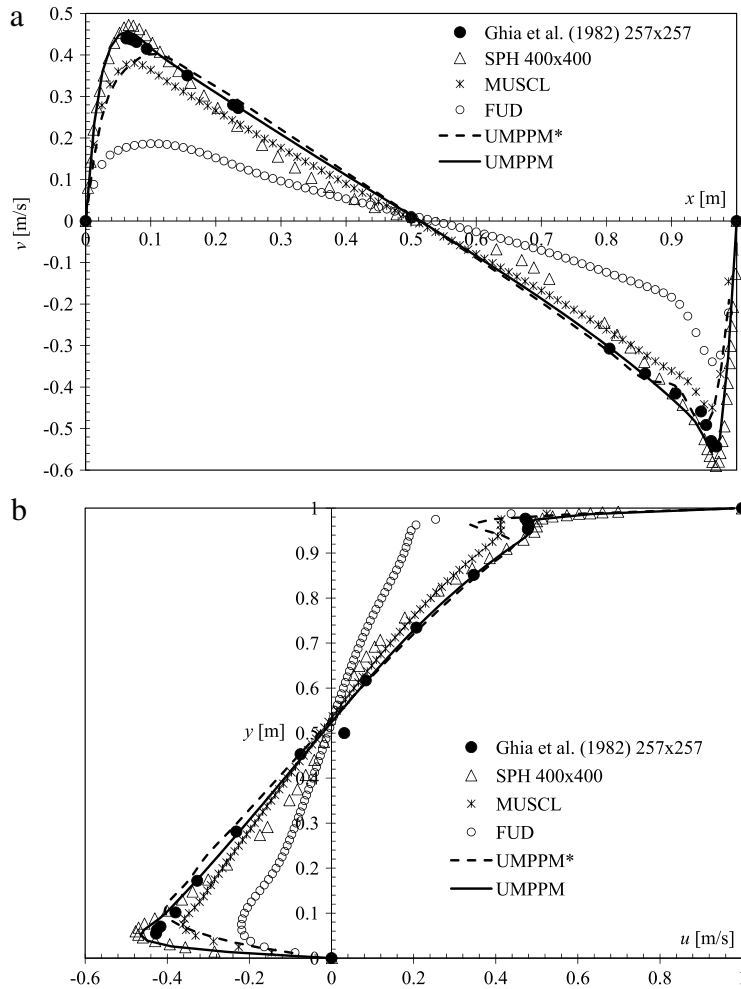


Fig. 14. Velocities at the (a) horizontal and (b) vertical centerlines of the square cavity. $Re = 10\,000$. For case UMPPM, $\alpha = 0.09$.

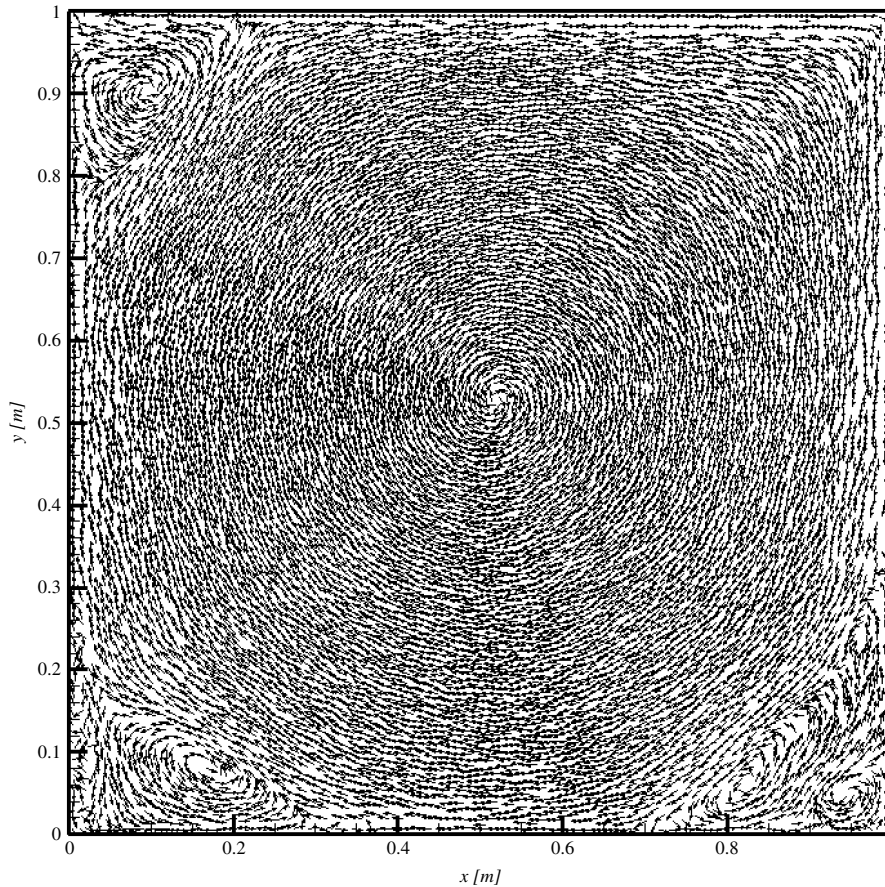


Fig. 15. Velocity vectors resulted from the lid-driven flow in the square cavity ($Re = 10\,000$) at $t = 1000$ s. The length of the vector is *not* scaled to the particle's speed to reveal clearly the secondary vortices near the corners. Case UMPPM of pressure mesh 80×80 is presented. $\alpha = 0.09$.

Step 6: Solve the intermediate particle velocity from Eq. (14). The diffusion term is treated implicitly via the procedure outlined in Section 3.2.1. At this stage, if heat transfer problem is considered, the source term (e.g. Eq. (4)) can be included in the momentum equation. Subsequently, the energy equation (Eq. (3)) is solved implicitly as:

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{\rho_i C_{p,i}} k_i \nabla^2 (T_i)^{n+1}. \quad (37)$$

Step 7: Advect the fluid particles to their intermediate positions \vec{r}_i^* in accordance with Eq. (9).

Step 8: Interpolate the intermediate velocities at mesh face f and mesh centroid P using the Moving Least Square (MLS) procedure outlined in Eq. (32). Compute the source term of the PPE accordingly by using the intermediate face velocities (RHS of Eq. (20)).

Step 9: Solve the PPE to obtain the new pressure field P_p^{n+1} .

Step 10: Obtain the new velocity and position of particle i by Eqs. (7) and (10), respectively. Accordingly, the new velocity of mesh P can be calculated by Eq. (36). Based on the corrected positions of the fluid particle, the temperature of mesh P (T_p^{n+1}) is interpolated from the temperature values stored at the neighboring fluid particles via the MLS method.

Step 11: Particles are deleted/added as necessary as discussed in Section 3.2.2.2.

Step 12: Go back to Step 5 and repeat the process until the desired time level is reached.

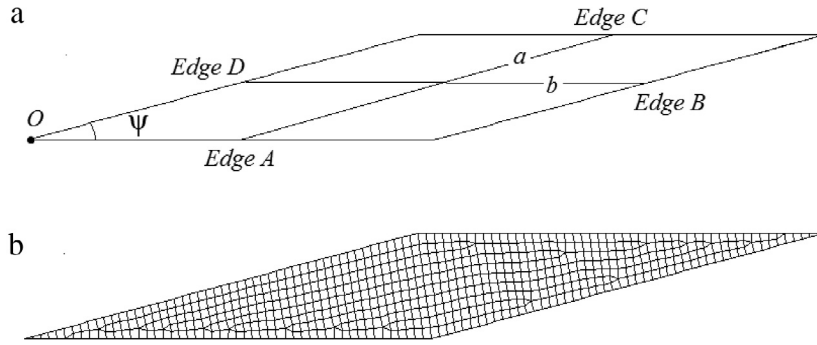


Fig. 16. Lid-driven flow in a skewed cavity. (a) Schematic diagram. The cavity is having a uniform side length $L = 1.0$ m. Lines a and b are connecting the midpoints of Edges A & C and Edges B & D , respectively. Point O is the origin $(0, 0)$. (b) Pressure mesh ($N_m = 589$, $l_{avg} = 0.02$ m).

4. Result and discussion

Here, the solution obtained from the particle deletion (i.e. depending on the α value) and particle addition procedures outlined in Section 3.2.2.2 is denoted as UMPPM in this section. If such procedure is employed, as we shall witness later, the total number of particles N_p is likely to increase as time progresses. This may contribute to higher flow accuracy at the expense of higher computational cost. In the current study, we have further investigated the possibility whereby N_p is enforced to be similar to N_m (i.e. number of pressure meshes). This is achievable by retaining only the fluid particle that is closest to the mesh centroid, while deleting the other fluid particles lying on the same pressure mesh. The solution obtained from this approach is named as UMPPM* in this section.

4.1. Taylor–Green flow

The Taylor–Green flow consists of a series of decaying vortices in the x – y plane ($x \in [0, 1]$; $y \in [0, 1]$) in which the flow field can be expressed analytically as:

$$u_{theo}(x, y, t) = -Ue^{bt} \cos(2\pi x) \sin(2\pi y) \quad (38)$$

$$v_{theo}(x, y, t) = Ue^{bt} \sin(2\pi x) \cos(2\pi y) \quad (39)$$

$$P_{theo}(x, y, t) = -\frac{U^2}{4} e^{2bt} (\cos(4\pi x) + \cos(4\pi y)). \quad (40)$$

Here, b is the decaying rate defined as $-8\pi^2/Re$ and the Reynolds number Re ($= \rho UL/\mu$) is prescribed as 100. The reference speed U , density ρ and domain length L are treated as unity in this case, similar to those reported in the previous SPH works [58,59].

The flow field is initialized from the analytical solution given above. Regarding the space discretization of the flow field, the unit square domain is firstly discretized by the square elements, followed by decomposition of each square element into four triangular elements (hence pressure mesh) as shown in Fig. 7. The total number of pressure meshes generated is denoted as N_m . Simulation is executed until $t_{max} = 5$ s (with CFL = 0.25, similar to that adopted by Ellero et al. [58] and Adami et al. [59]). And, in order to check the convergence of the current method, the total number of particles N_p is enforced to be similar as the total number of pressure meshes N_m (i.e. the UMPPM* solution described in Section 4).

Fig. 8 shows the instantaneous particle speed and fluid pressure at $t = 2$ s for $N_m = 2500$. Velocity vectors are plotted as well to reveal the flow structure. It is interesting to observe the smoothness of velocity and pressure field as we do not employ any artificial numerical treatments in our UMPPM* method. The decay of the maximum speed over time is shown in Fig. 9. It is encouraging to see that the agreement is very good even though a coarse particle resolution is considered ($N_m = 1600$). As compared to the ISPH result ($N_p = 1600$) reported by Ellero et al. [58] and the recent Weakly Compressible SPH (WCSPH) result ($N_p = 2500$) reported by Adami et al. [59], our UMPPM*

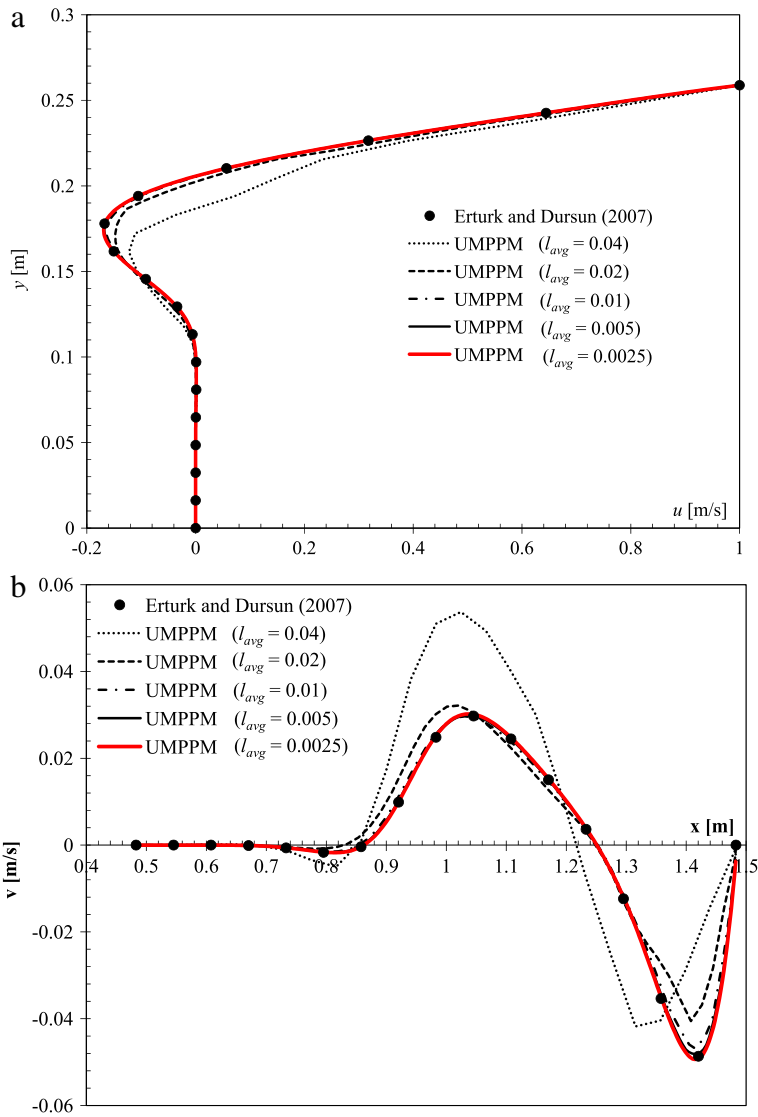


Fig. 17. Predicted velocities at $t = 10$ s along (a) line a and (b) line b . For case UMPPM, $\alpha = 0.05$.

solution is considerably smoother and closer to the theoretical solution. In order to quantify the numerical error, we have adopted the relative error L_∞ (similar to Adami et al. [59]), defined as:

$$L_\infty(t) = \frac{|\max \|\bar{u}_P\| - Ue^{bt}|}{Ue^{bt}}. \tag{41}$$

Fig. 10 shows the relative errors of maximum speed obtained at different particle resolutions as time progresses. The order of convergence of the UMPPM* method is almost comparable to that of the ISPH method reported by Ellero et al. [58] as reported in Fig. 11, while the error level of the UMPPM* method is lower than that of the ISPH method. As shown in Fig. 10, the error levels are considerably smoother as compared to those of the WCSPH computation reported by Adami et al. [59]. Also, it is interesting to note that the relative error (case of coarsest resolution, i.e. $N = 1600$) is below 2%, which is considerably lower than the numerical errors (fluctuating within 5%) of all the test cases reported in the literature (cf. Figure 8(b) in [59]).

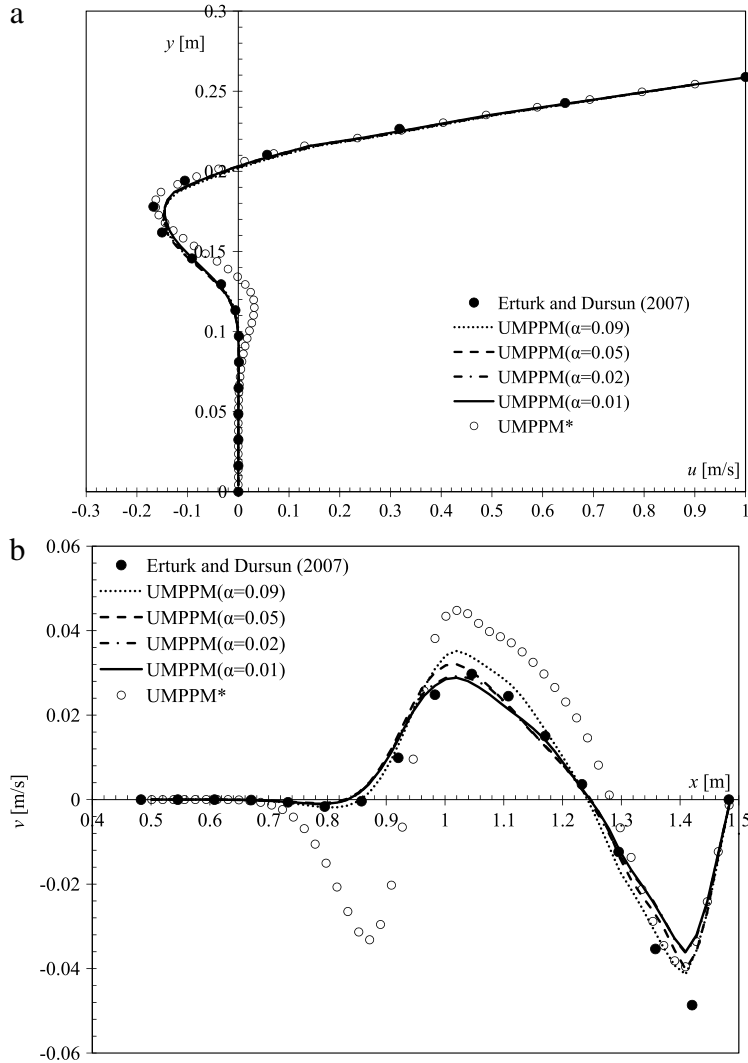


Fig. 18. Predicted velocity profiles at $t = 10$ s by for UMPPM (using different α value shown in bracket) and UMPPM* solutions by using an average pressure mesh size of $l_{avg} = 0.02$ m on (a) line a and (b) line b .

As the last check of the spatial convergence, the L_2 norm of the x -velocity field is computed as:

$$\|u\|_2 = \sqrt{\frac{\sum_N (u(t) - u_{theo}(t))^2}{N_m}}, \tag{42}$$

and the result is shown in Fig. 12. The error is reduced progressively as the number of particles is increased with the order of convergence ~ 1 .

4.2. Lid-driven flow

4.2.1. Square cavity

We consider the lid-driven flow in a square cavity at high Reynolds number ($Re = 10000$). This is a computationally challenging problem as strong velocity gradient exists near the walls and several secondary vortices exist at the corner. The square cavity with side length $L = 1.0$ m is filled with a fluid of density $\rho = 1.0$ kg/m³ and $\mu = 1/Re$ Pa s. Here, the top wall is moving at a constant x -velocity, i.e. $u = 1.0$ m/s.

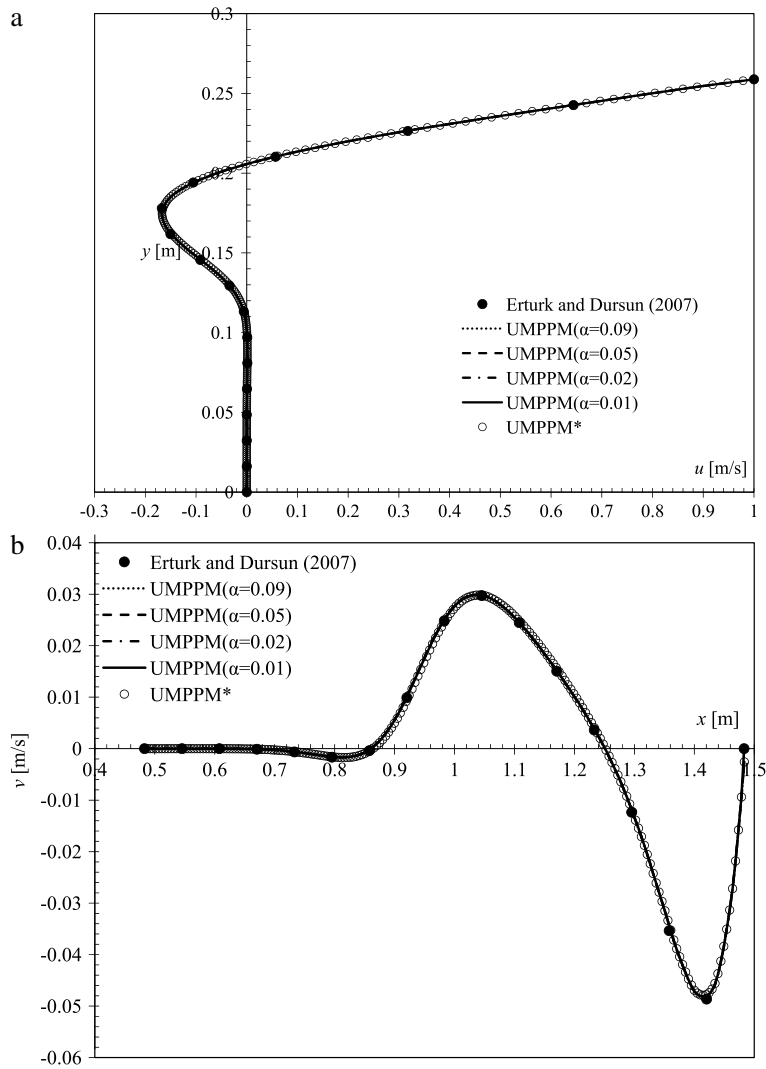


Fig. 19. Predicted velocity profiles at $t = 10$ s by UMPPM (using different α value shown in bracket) and UMPPM* solutions by using an average pressure mesh size l_{avg} of 0.005 m on (a) line a and (b) line b .

At such a high Re flow condition, it is interesting to note from the original MPPM work by Hwang [26] that the flow profiles at the mid-sections of the square cavity can be represented well (see Figure 14 in [26]) even on a coarse pressure mesh resolution of 80×80 (or $N_m = 6400$). Owing to the adaptive nature of the original MPPM method (particle addition/deletion was performed, $\alpha = 0.09$), the total number of particles may become larger than the total number of pressure meshes employed (see Fig. 13). As one shall notice later, this feature is particularly useful in enhancing the flow accuracy at a reasonable number of pressure meshes.

Similar to the original MPPM work, the simulation is executed until $t = 1000$ s ($CFL = 0.1$, similar to Hwang [26]) to ensure a statistically steady solution on the 80×80 square pressure mesh. As reported in Fig. 14, it is interesting to note that the adaptive UMPPM solution is already close to the fine grid solution of Ghia et al. [60]. For this case, the number of particles $N_p/N_m \sim 2.3$ at steady-state condition as reported in Fig. 13. Without particle adaptation (i.e. case UMPPM*), seemingly, the high velocity gradients near the wall are under-predicted and the insufficient particle resolution may yield an unphysical overshoot of velocity profile near the top wall. By using the same mesh resolution (80×80), the finite-volume solutions are obtained as well from the commercial code (FLUENT). Here, two convective discretization schemes, namely the First-order Upwind Differencing (FUD) scheme and the high-resolution MUSCL scheme are considered. It is appealing to note that the numerical solutions of UMPPM* (no

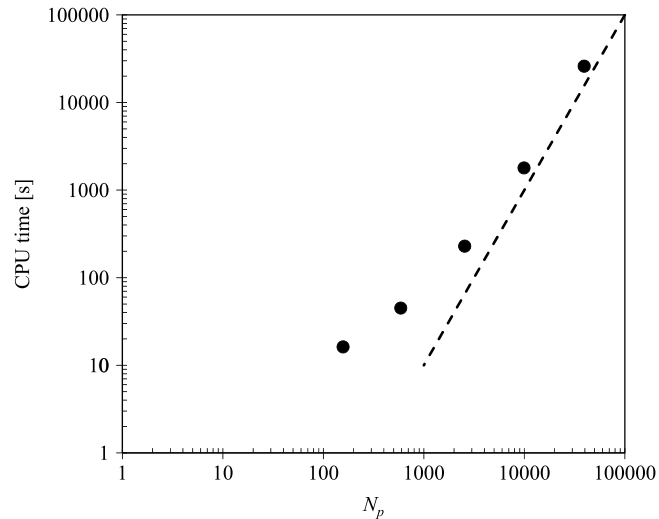


Fig. 20. CPU time versus number of particles N_p for the case of lid-driven flow in the skewed cavity by using UMPPM* solution method.

Table 1
CPU time taken for different pressure mesh sizes l_{avg} . For case UMPPM, $\alpha = 0.05$, CFL = 0.5.

l_{avg} (m)	CPU time (s)	
	UMPPM*	UMPPM
0.04	16.22	18.02
	157 ^a	199 ^a
0.02	45.04	53.63
	589 ^a	825 ^a
0.01	229.17	328.64
	2549 ^a	3964 ^a
0.005	1792.86	2557.11
	9920 ^a	16 802 ^a
0.0025	25 987.41	35 355.43
	39 395 ^a	71 246 ^a

Note: Computations are performed on Intel (R) Xeon (R) CPU E5-1640V2@3.7 GHz (8CPU) using a serial C code.

^a Indicates the number of particles at $t = 10$ s.

adaptation) exhibit less smearing as compared to the solutions of high-resolution MUSCL scheme. As expected, the excessive numerical diffusion in the FUD scheme is very prominent at this high Re flow.

On the other hand, Adami et al. [59] have recently attempted their newly developed SPH technique in this lid-driven problem as well. The SPH results obtained by using their finest particle resolution (400×400) are reported in Fig. 14. While the high velocity gradients in the vicinity of the walls are nicely captured, their numerical solutions at the core region are somehow not agreeable with the reference solutions of Ghia et al. [60]. As reported by Adami et al. [59], the primary vortex is more centered with increasing Re , which is observed as well in our current study (see Fig. 15). Also, it is appealing to note that the secondary vortices are nicely captured in the corners of the square cavity even though the resolution employed in the current UMPPM method is considerably coarser as compared to the one used by Ghia et al. [60] and Adami et al. [59].

4.2.2. Skewed cavity

Next, the lid-driven flow in a skewed cavity is considered. This is indeed the case computed previously by Erturk and Dursun [61] whereby they have solved the flow equations by using the vorticity–stream function approach on a very fine grid resolution of 512×512 . Similar to the lid-driven flow in the square cavity, the side length L is prescribed

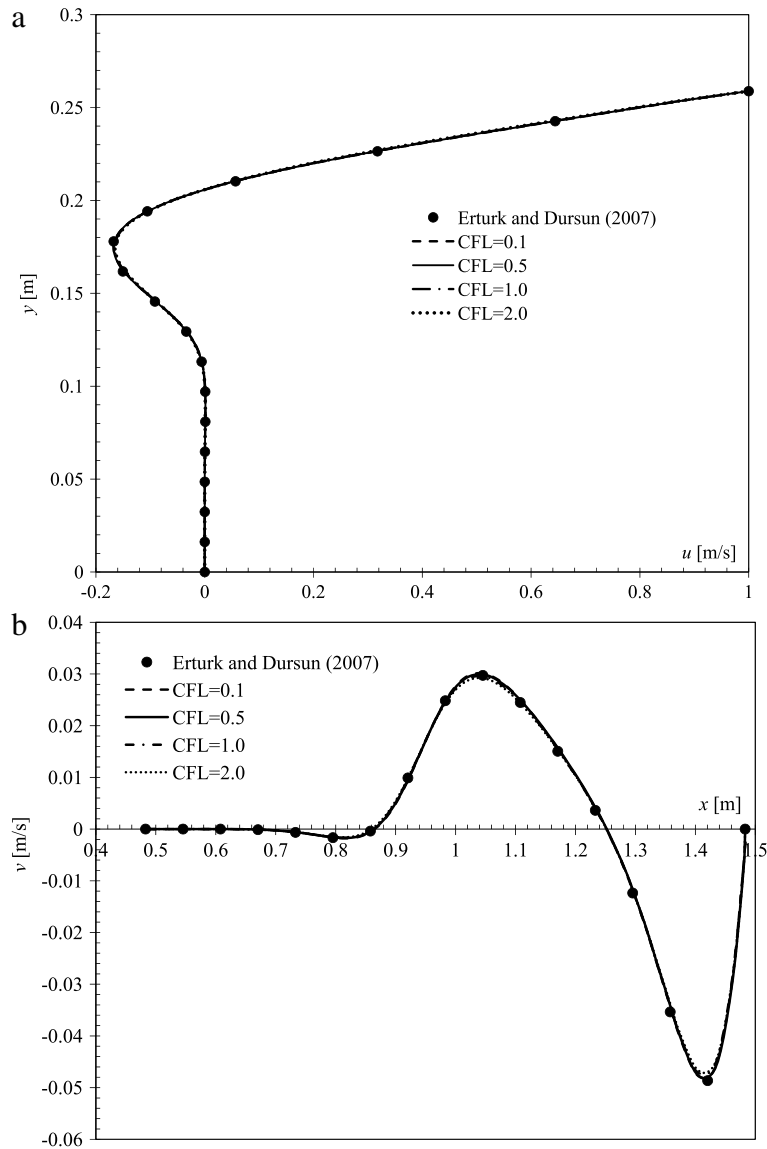


Fig. 21. The effect of CFL number on the predicted velocity profiles at $t = 10$ s by using UMPPM* method on average pressure mesh size $l_{avg} = 0.005$ m on (a) line a and (b) line b .

as 1.0 m. The fluid density ρ is 1.0 kg/m^3 and fluid dynamic viscosity μ is 0.001 Pa s . Again, the top wall is moving at a constant x -velocity $u = 1.0 \text{ m/s}$ (hence $Re = 1000$). Here, the skewed angle ψ is 15° , which is similar to the case of having the most extreme skewed angle reported by Erturk and Dursun [61]. The schematic diagram of the skewed cavity is shown in Fig. 16(a). In the current computational study, a hybrid pressure mesh system is employed. The layout of the pressure mesh (average mesh size $l_{avg} = 0.02 \text{ m}$) is shown in Fig. 16(b).

For each computation considered here, the simulation is executed until $t = 10$ s to ensure a statistically stationary solution. The velocity plots predicted by using the UMPPM method along line a and line b are shown in Fig. 17. Here, α is taken as 0.05. As seen, the numerical results are converging to the benchmark solutions of Erturk and Dursun [61] as the pressure mesh (hence particle resolution) is refined. Seemingly, the average mesh size of 0.01 m is adequate enough to resolve the u -velocity along line a . However, along line b , a finer mesh resolution ($l_{avg} = 0.0025 \text{ m}$, $N_m = 39\,395$, $N_p = 71\,246$ at $t = 10$ s) is needed to resolve the relatively high velocity gradient in the vicinity of Edge B . It is worth to mention here that the numerical results obtained from this particular mesh resolution

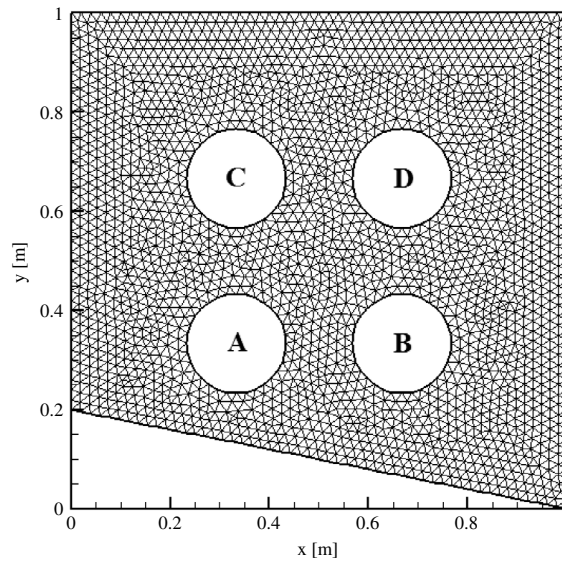


Fig. 22. Unstructured pressure mesh embedded in a complex cavity consisting of 4 cylinders *A*, *B*, *C* and *D*. The centers for cylinders *A*, *B*, *C* and *D* are $(1/3, 1/3)$, $(2/3, 1/3)$, $(1/3, 2/3)$ and $(2/3, 2/3)$, respectively. The top boundary is subjected to a constant x -velocity of 1 m/s.

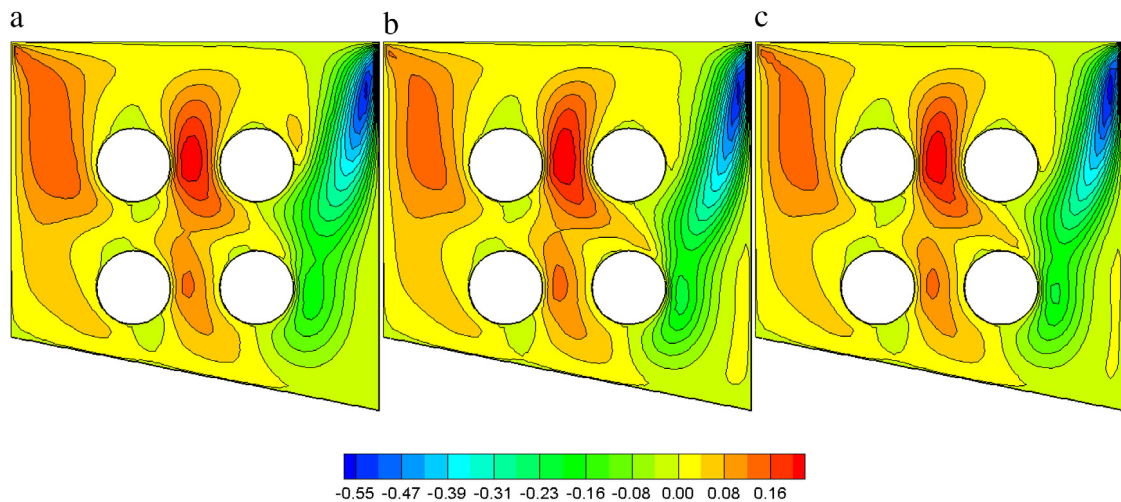


Fig. 23. Comparison of y -velocity contour obtained from (a) FUD scheme (FLUENT). (b) MUSCL scheme (FLUENT) and (c) UMPPM* scheme. The plotted y -velocity is ranging between -0.55 m/s and 0.2 m/s. Within the considered range, 20 contour lines are plotted.

are almost similar to the benchmark solutions of Erturk and Dursun [61] obtained on the very fine mesh system (i.e. 512×512).

In fact, the effectiveness of employing the particle adaptation is more prominent when the resolution of the pressure mesh is coarse. As observed from Fig. 18 for the case of $l_{avg} = 0.02$ m, the UMPPM* solution is not comparable at all with the reference one. Interestingly, upon implementing the particle adaptation procedure (UMPPM), the numerical solutions are now closer to the reference data as shown in Fig. 18. However, as the resolution of the pressure mesh is refined, the effectiveness of the particle adaptation strategy is no longer apparent as shown in Fig. 19. Table 1 reports on the CPU time needed to complete the flow simulation until $t = 10$ s. For the case of UMPPM*, the increase in CPU time is approaching $O(N_p^2)$ as N_p is increased as shown in Fig. 20.

One of the attractive components of the current method is its implicit feature which has enabled us to adopt a larger time step size for numerical computation. As reported in Fig. 21, there is no significant discrepancy between the flow

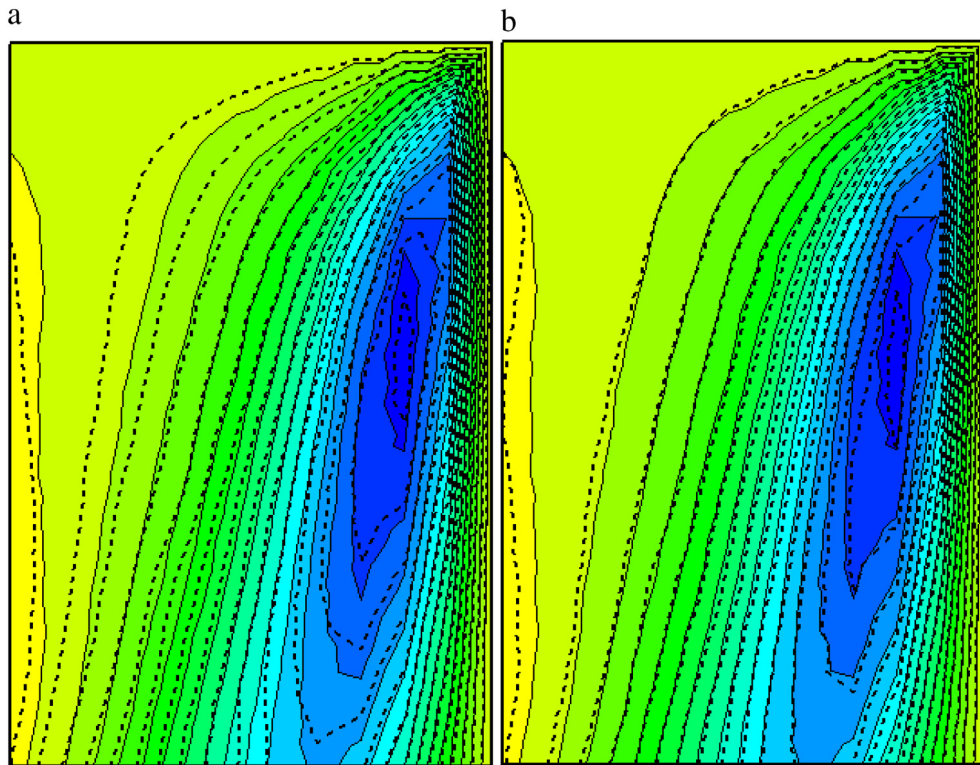


Fig. 24. Overlay of the y -velocity contour lines (dashed-lines) predicted from (a) first-order upwind scheme (FLUENT) and (b) third-order MUSCL scheme (FLUENT) on the UMPPM* solution (solid contour lines).

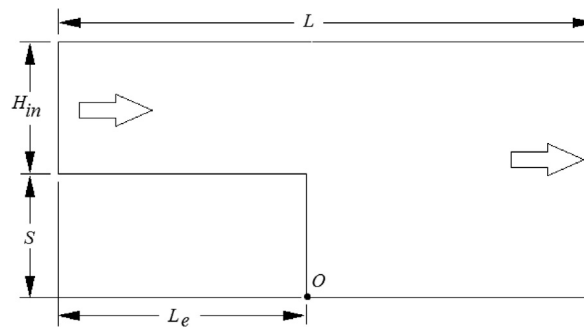


Fig. 25. Schematic diagram of the backward-facing step. $H_{in} = 5.2$ m; $S = 4.9$ m; $L_e = 32.6$ m and $L = 122.4$ m. Point O is the origin.

solutions adopting different CFL numbers, except for the fact that there is a slight under-prediction of the under- and over-shoots of the velocity profiles if CFL is > 1.0 .

4.2.3. Complex cavity

In order to demonstrate the capability of the current method in handling flow involving complex geometry, a lid-driven flow in a complex cavity such as that illustrated in Fig. 22 is considered. An unstructured pressure mesh with an averaged mesh size of 0.02 m is adopted in this flow case. The Reynolds number defined as: $Re = UL/\nu$ is 1000. Here, U is the speed of the moving lid (top boundary), prescribed as 1.0 m/s and L is the length of the top boundary, i.e. $L = 1.0$ m (see Fig. 22). The UMPPM* results are compared with those obtained from the finite-volume method (FLUENT) as well. Here, it is important to note that a similar mesh is used in FLUENT while the comparison of flow solutions is made.

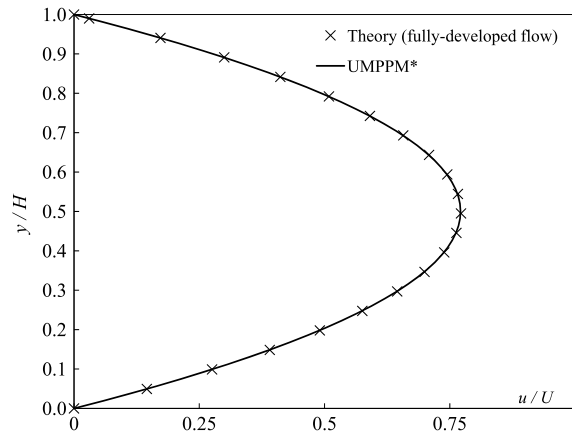


Fig. 26. Comparison of fully-developed velocity profiles. UMPPM* solutions are obtained at section $x/S = 17.84$. $H = H_{in} + S = 10.1$ m. $U = 0.14$ m/s.

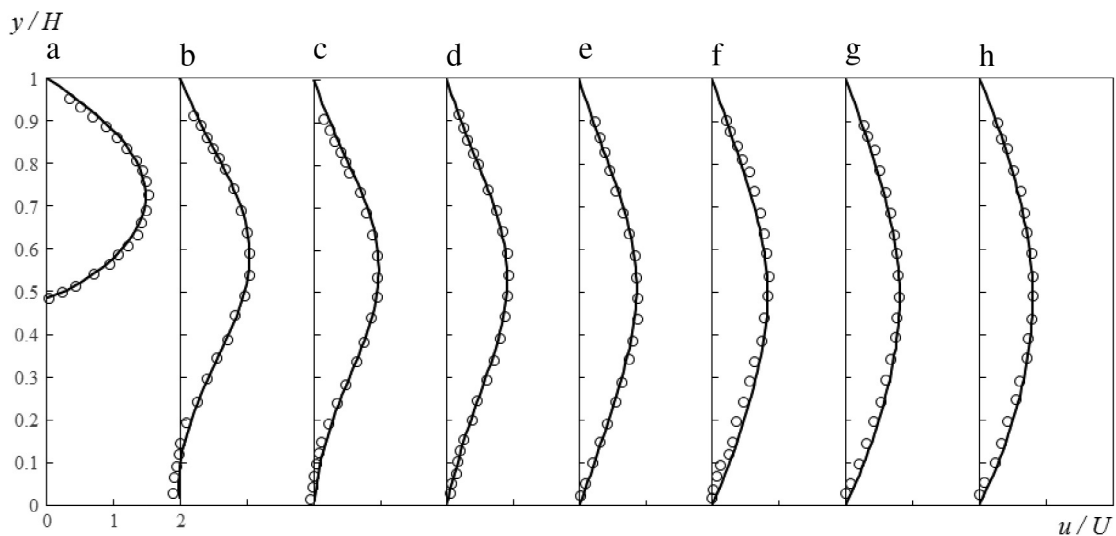


Fig. 27. Comparison of velocity profiles at (a) $x/S = 0$; (b) $x/S = 2.55$; (c) $x/S = 3.06$; (d) $x/S = 3.57$; (e) $x/S = 4.18$; (f) $x/S = 4.80$; (g) $x/S = 5.41$ and (h) $x/S = 6.12$. Empty circle: experimental data [62]; Solid line: UMPPM* solution. $H = H_{in} + S = 10.1$ m. $U = 0.14$ m/s.

Fig. 23 shows the contour lines of the y -velocity values obtained from the finite volume solutions and the UMPPM* solution. Different convective discretization schemes are considered during the execution of the finite-volume computation, i.e. the First-order Upwind Differencing (FUD) scheme and the third-order MUSCL scheme. As seen, three sets of numerical results are comparable to each other. The ascending jet between the top cylinders is well captured by the UMPPM* method. In order to better reveal the capability of UMPPM* in handling flow involving sharp gradient, a close-up view of the descending jet which is observed in the vicinity of top right corner of the cavity, is plotted in Fig. 24. From these plots, it is apparent that our UMPPM* solution is coming closer to that of the high-resolution MUSCL scheme. The smearing of the velocity field predicted by the FUD scheme (Fig. 24(a)) is rather obvious.

4.3. Inflow–outflow problem

4.3.1. Flow over a backward-facing step

Here, the method is used to solve the flow over a backward-facing step as illustrated in Fig. 25. This case has been previously computed by Adami et al. [59] by using the SPH method, whereby they have imposed a constant velocity flow condition ($U = 0.14$ m/s) at the inlet. In order to ensure a fully-developed flow condition at the outlet, the total

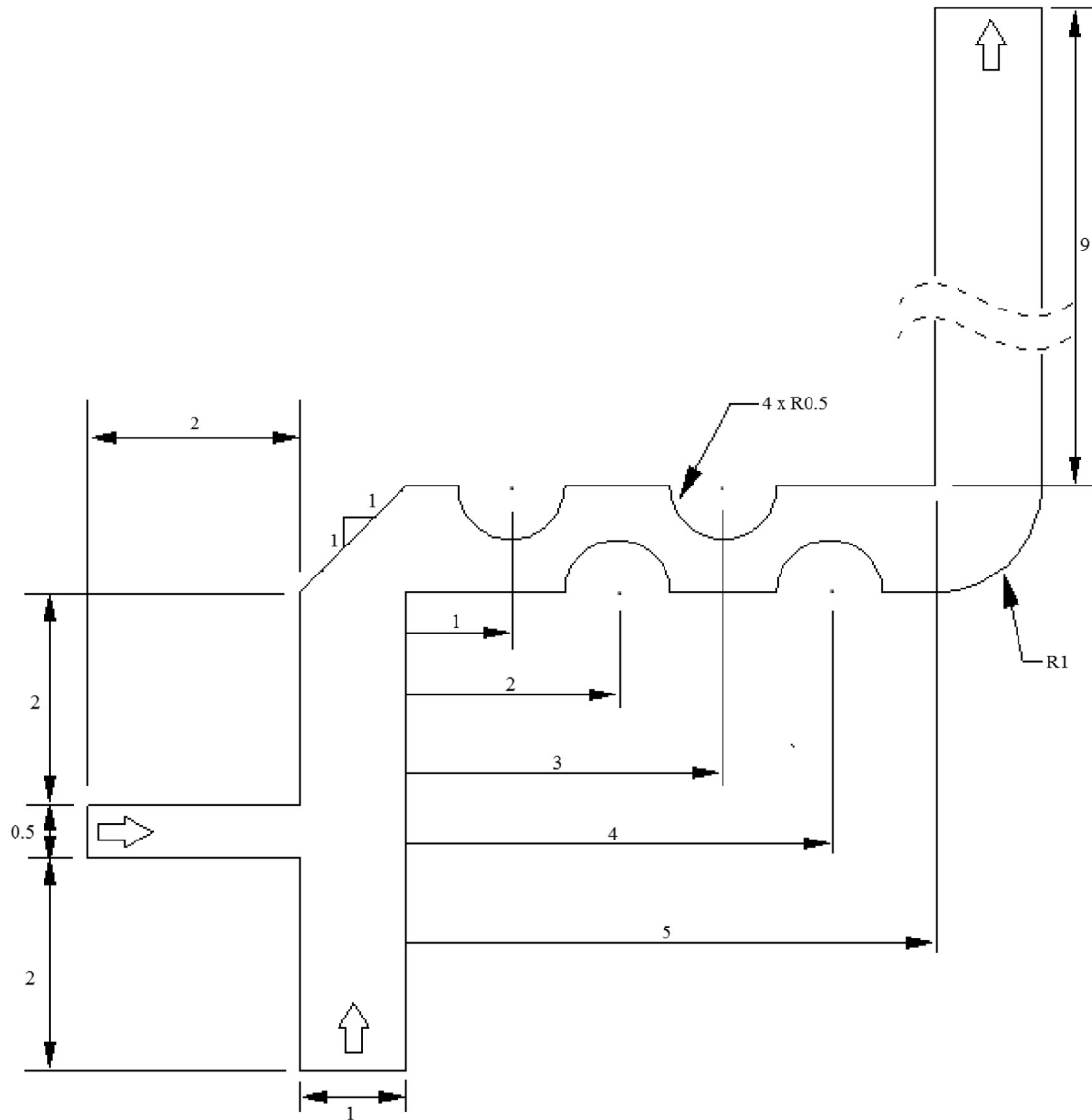


Fig. 28. Complex flow channel involving 2 inlets and 1 outlet. All dimensions are in (m).

length of the flow domain is prescribed as $L = 122.4$ m. A pressure mesh with an average size of 0.2 m is used to discretize on the flow domain. The fluid having density of 1.0 kg/m^3 and dynamic viscosity of 0.01456 Pa s is considered here ($Re = 2\rho U H_{in}/\mu = 100$).

By using the above numerical settings, the simulation has been executed until $t = 2000$ s to ensure a statistically steady result. The predicted fully-developed flow profile at $x/S = 17.84$ is shown in Fig. 26. As seen, the UMPPM* solution compares very well with the theoretical solution. Also, the velocity profiles at different flow sections are depicted in Fig. 27 and the numerical solutions exhibit good agreement with the experimental data of Armaly et al. [62].

4.3.2. Flow in a complex channel

This flow case involves multiple inlets and one outlet, as illustrated in Fig. 28. There is an inflow from the bottom opening of the primary flow channel of width $W = 1.0$ m, whereby the fluid particles will undergo a meandering

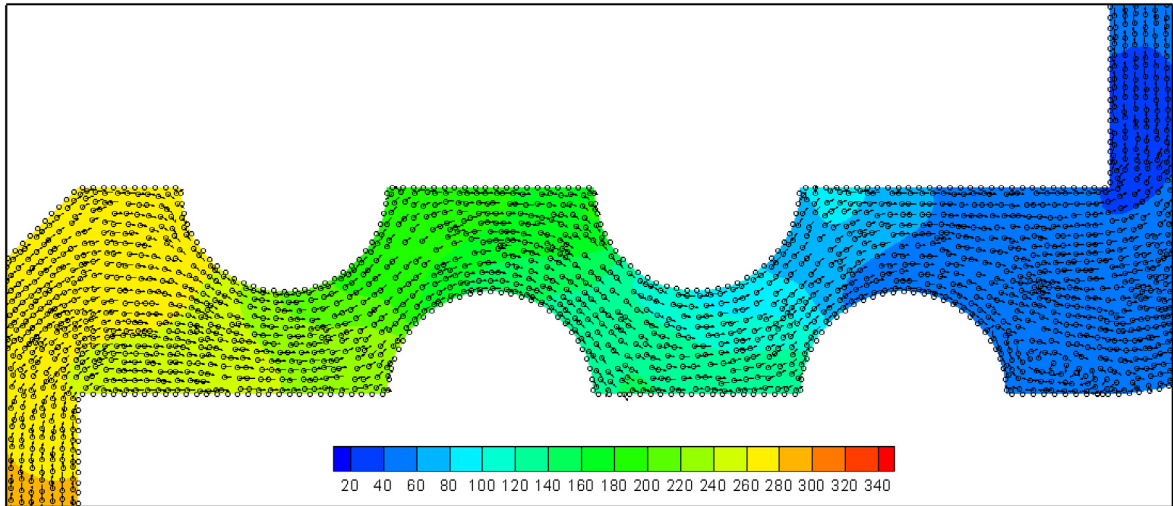


Fig. 29. Movements of flow particles in the vicinity of the semi-circular grooves at $t = 10$ s. Velocity vectors are *not* scaled to their speed value. Pressure drop (Pa) across the grooved channel is indicated in the contour plot.

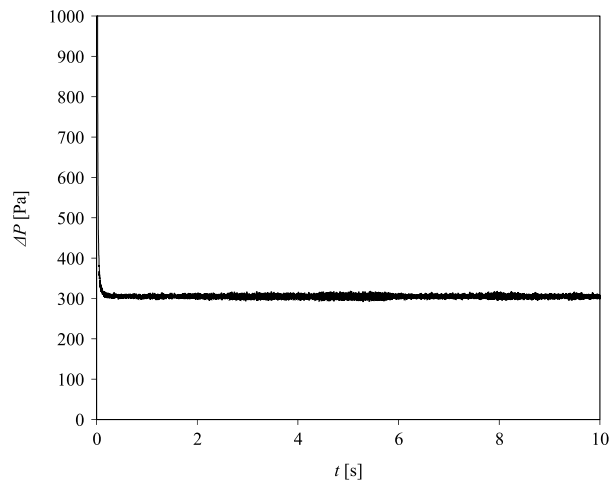


Fig. 30. The pressure drop between the bottom inlet and outlet predicted from UMPPM* method.

motion (due to the semi-circular grooves) before exiting the outlet located at the top. Prior to reaching the grooved channel, the fluid particles in the primary flow channel are mixed with those from the secondary flow channel (with smaller width $W_s = 0.5$ m) as reported in Fig. 28. In this case, the primary and the secondary inflow speeds are 1.456 m/s and 2.912 m/s, respectively. Both inflow directions are normal to the corresponding flow boundaries. The Reynolds number ($Re = \rho U W / \mu$) is 10, where ρ is 1.0 kg/m³, μ is 0.1456 Pas, U is the primary inflow speed ($= 1.456$ m/s).

The UMPPM* computation is performed on a pressure mesh with an average mesh size of 0.05 m. There are four semi-circular grooves in the middle section of the complex flow channel, which will contribute to significant pressure drop of the fluid particles as delineated in Fig. 29. The meandering motion of the fluid particles is quite visible. At such a low Re flow condition, flow separations are hardly found near the boundaries of the semi-circular grooves, except that a visible recirculating flow region can be found behind the last semi-circular groove. The pressure difference between the outlet and the primary flow inlet (bottom opening) during the progress of the simulation is reported in Fig. 30. The time-averaged pressure difference (taken between 2 and 10 s) is ~ 304.58 Pa.

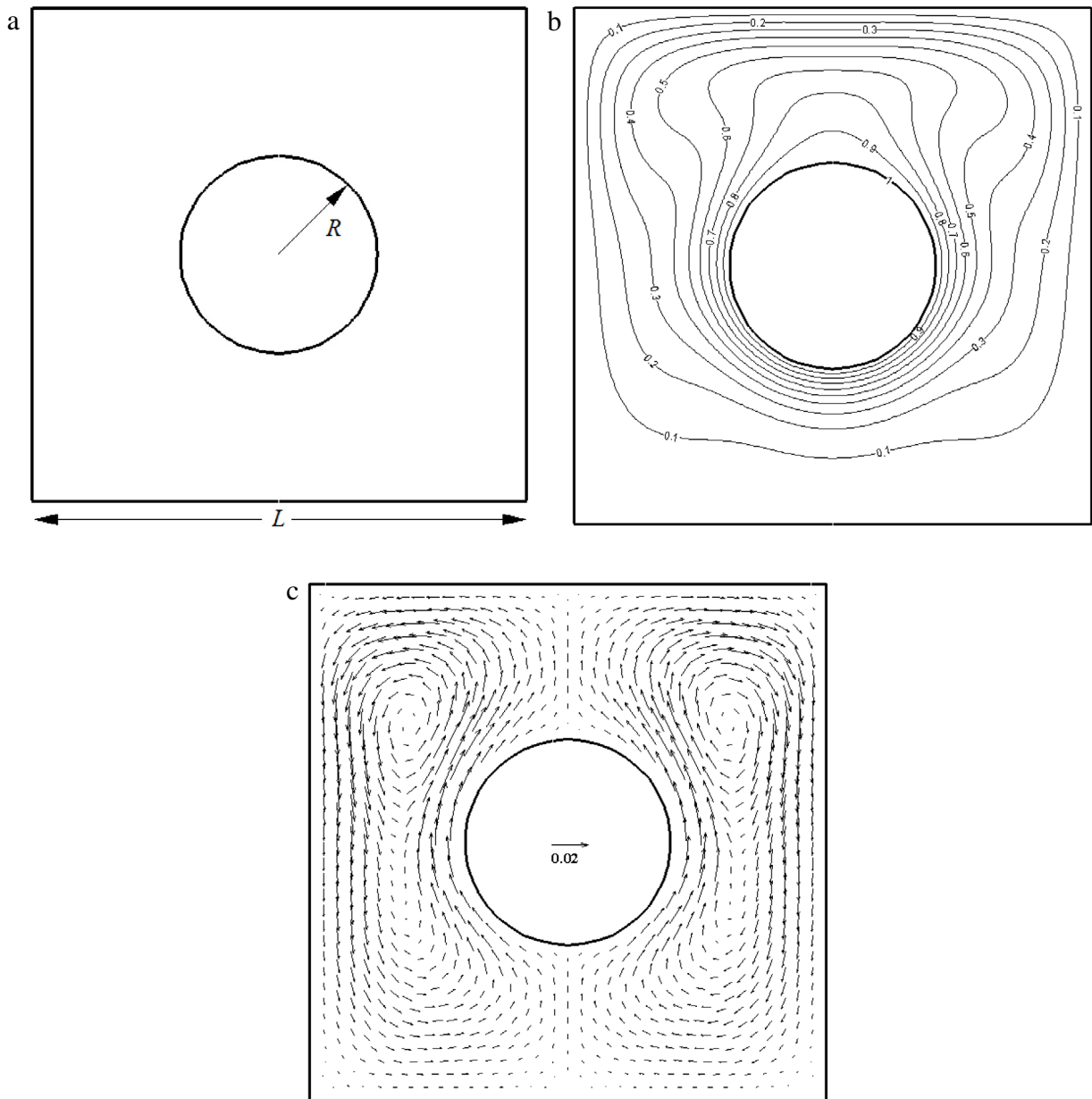


Fig. 31. Natural convection problem in a concentric annulus. (a) Schematic diagram of the flow domain. (b) Dimensionless temperature contour plot $T^* = (T - T_1)/(T_1 - T_2)$. (c) Velocity vector. UMPPM* method ($N_m = 2156$) is used. $Ra = 1 \times 10^5$.

4.4. Natural convection

4.4.1. Concentric annulus between an inner circular cylinder and an outer square cylinder

Next, the current method is used to study the natural convective heat transfer between a concentric annulus as shown in Fig. 31(a). Here, R and L are prescribed as 0.02 and 0.10 m, respectively which translate to parameter $rr = L/2R = 2.5$ as defined in [63]. The temperatures of the inner circular wall and outer square wall are fixed at $T_1 = 301.5518$ K and $T_2 = 300$ K, respectively. Here, the reference temperature T_o that appeared in the buoyancy force term (Eq. (4)) is prescribed as T_2 . With the Rayleigh number ($Ra = \rho^2 C_p g \beta (T_1 - T_2) D_1^3 / k \mu$) of 1×10^5 , a hot rising stream can be seen in the vicinity of the inner hot circular cylinder as shown in Fig. 31(b), (c). The working

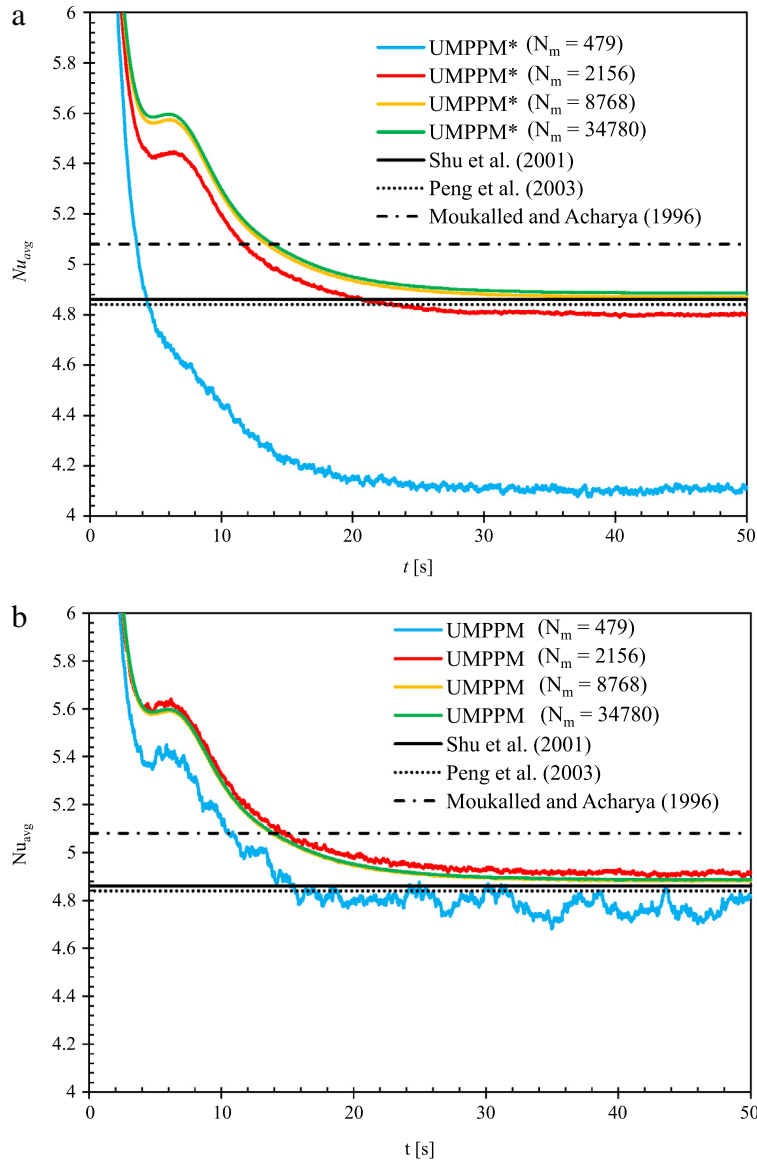


Fig. 32. Time evolution of Nu_{avg} using (a) UMPPM* method and (b) UMPPM method ($\alpha = 0.05$). The reference solutions of Moukalled and Acharya [65], Shu et al. [63] and Peng et al. [64] are 5.08, 4.86 and 4.84, respectively.

fluid with the following properties: specific heat $C_p = 1006.3$ J/kg K, thermal conductivity $k = 0.02816$ W/mK, thermal expansion coefficient $\beta = 0.003$ K $^{-1}$, dynamic viscosity $\mu = 0.00002$ Pa s is considered. The gravitational acceleration is assumed as -10 m/s 2 .

The local heat transfer at the inner circular wall can be computed as:

$$q = h(T_1 - T_2) = -k \frac{\partial T}{\partial n}. \tag{43}$$

Therefore, the local convection coefficient h can be obtained:

$$h = -k \frac{\partial T}{\partial n} \frac{1}{(T_1 - T_2)}. \tag{44}$$

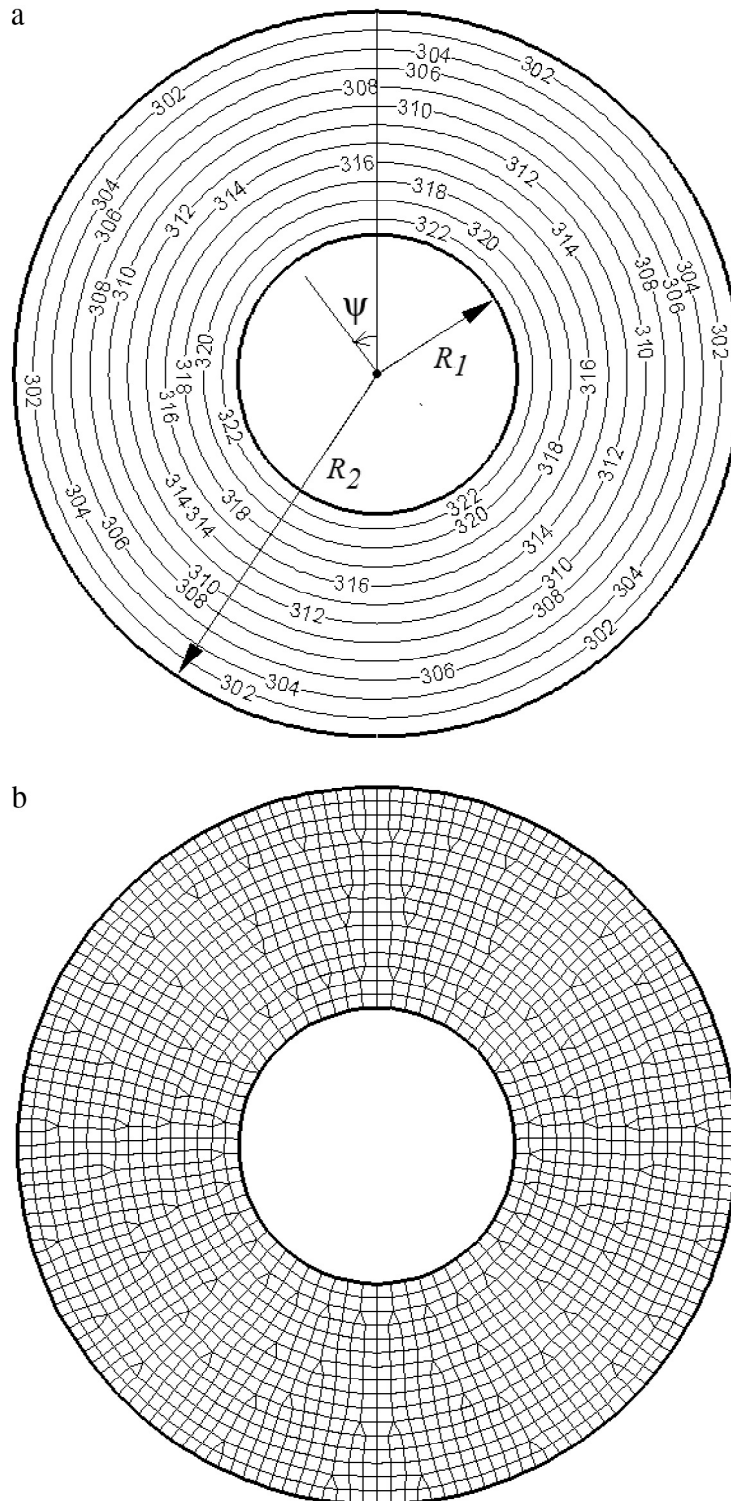


Fig. 33. Natural convection problem in two concentric cylinders. (a) Schematic diagram with linear temperature (K) profile at $t = 0$ s. (b) Hybrid mesh used in UMPPM* ($N_m = 1916$; $l_{avg} = 2$ mm).

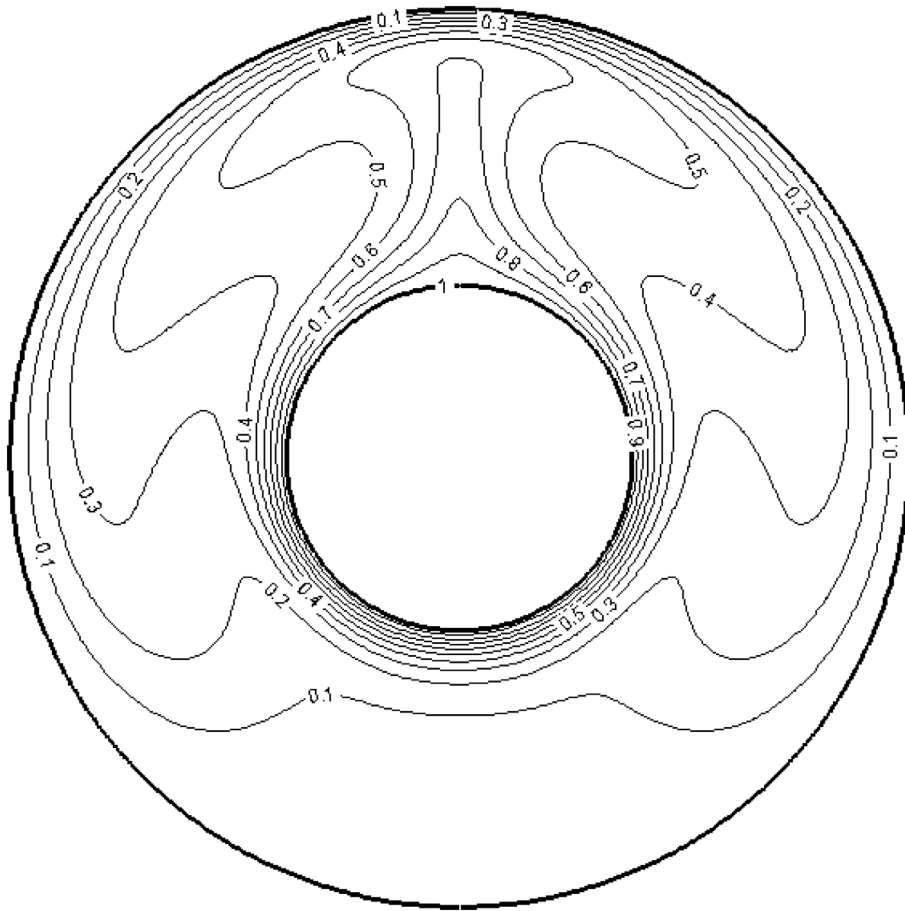


Fig. 34. Contour lines of dimensionless temperature $T^* = (T - T_1)/(T_1 - T_2)$. $N_m = 29656$ ($l_{avg} = 0.5$ mm). $Ra = 97600$.

During the computation, h is computed at each pressure mesh lying adjacent to the inner circular cylinder. The area-weighted average \bar{h} is then determined and consequently the average Nusselt number Nu_{avg} can be calculated as:

$$Nu_{avg} = \frac{\bar{h}S}{k}. \quad (45)$$

Here, S is taken as half of the circumferential length of the inner circular cylinder.

The time histories of Nu_{avg} are plotted in Fig. 32. By considering the coarsest pressure mesh resolution ($N_m = 479$), the numerical result obtained from the UMPPM* solution is not encouraging at all, as the predicted Nu_{avg} deviates considerably from the reference solutions as reported in Fig. 32(a). This shortcoming can be partly circumvented by adopting the UMPPM strategy (particle adaptation) as shown in Fig. 32(b), whereby the predicted Nu_{avg} (~ 4.8) is now coming closer to the reference solutions. As the pressure mesh is refined, the numerical solutions are approaching to the reference solutions. However, the grid-independent UMPPM and UMPPM* solutions are closer to those reported by Shu et al. [63] and Peng et al. [64].

4.4.2. Concentric annulus between two circular cylinders

Finally, the buoyancy-driven flow between the two concentric cylinders is considered. Fig. 33(a) illustrates the schematic diagram of two concentric cylinders with different radius ($R_1 = D_1/2 = 0.02$ m and $R_2 = 0.052$ m). Similar to Liang et al. [66], the flow temperature is initialized with a linear temperature profile (see Fig. 33(a)) at $t = 0$ s. The inner and outer temperatures of the cylindrical walls are prescribed as $T_1 = 323.664$ K and $T_2 = 300$ K, respectively. Again, the reference temperature T_o is taken as T_2 . Here, the fluid is considered as air with specific heat $C_p = 1006.3$ J/kg K, thermal conductivity $k = 0.02816$ W/mK, thermal expansion coefficient $\beta = 0.003$ K $^{-1}$ and

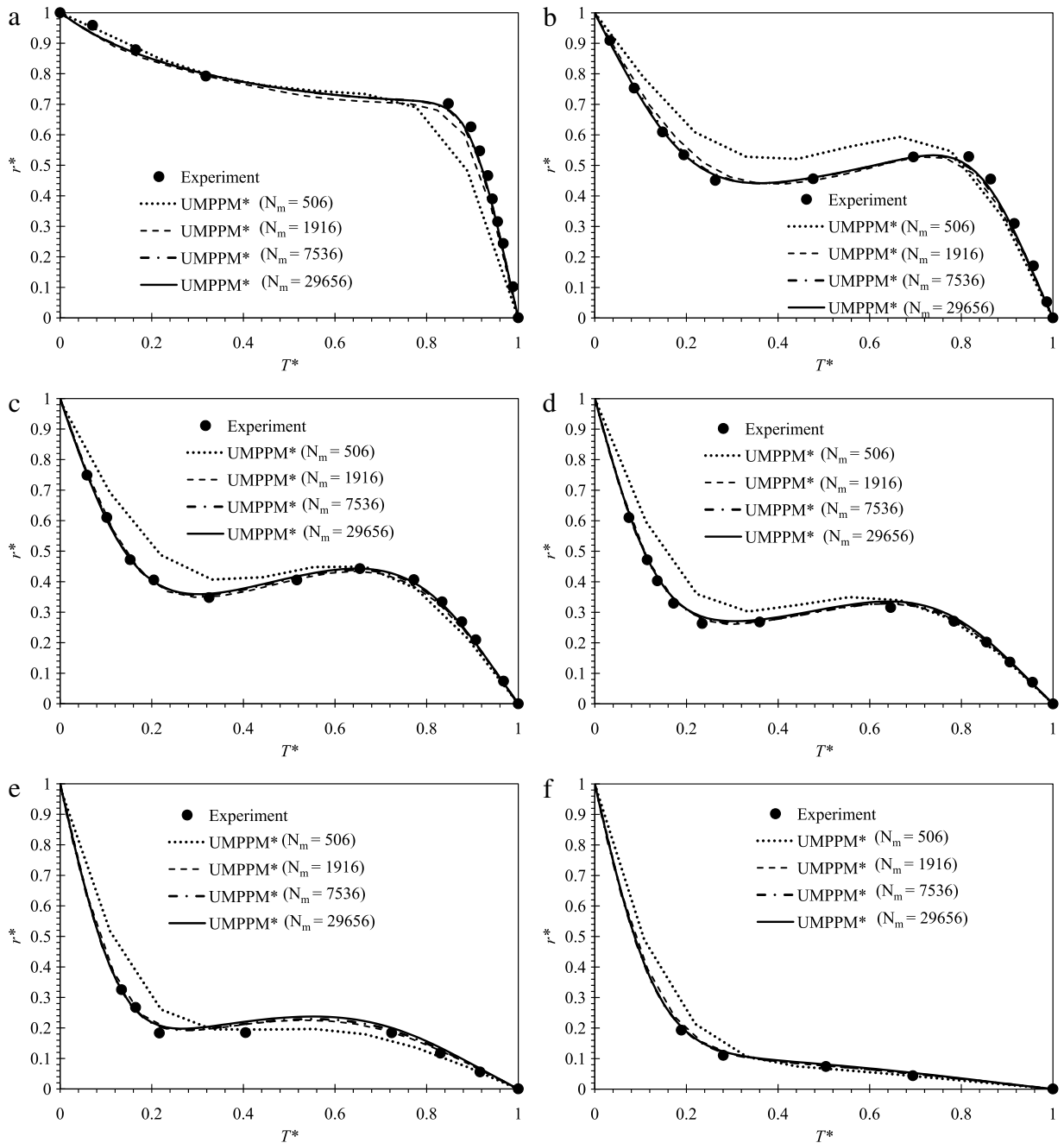


Fig. 35. Radial temperature profiles at different angular positions. (a) $\psi = 0^\circ$; (b) $\psi = 30^\circ$; (c) $\psi = 60^\circ$; (d) $\psi = 90^\circ$; (e) $\psi = 120^\circ$; (f) $\psi = 150^\circ$. Dimensionless radial distance and dimensionless temperature are $r^* = (R - R_1)/(R_1 - R_2)$ and $T^* = (T - T_1)/(T_1 - T_2)$.

dynamic viscosity $\mu = 0.00002$ Pa s. This translates into the Rayleigh number ($Ra = \rho^2 C_p g \beta (T_1 - T_2) D_1^3 / k \mu$) of 97 600, which is similar to the experimental condition of Kuehn and Goldstein [67]. The gravitational acceleration is assumed as -10 m/s².

The solutions are executed until $t = 20$ s to ensure a steady-state condition. Again, the computations are executed on a hybrid unstructured pressure mesh environment as shown in Fig. 33(b) for the case of $N_m = 1916$ ($l_{avg} = 2$ mm). Here, no particle adaptation is activated in the UMPPM* calculation. For this flow case of relatively high Ra , a thermal plume is expected to be seen on top of the inner cylinder as illustrated in Fig. 34. This is consistent with the very recent

MPS result presented by Liang et al. [66]. Fig. 35 shows the predicted temperature profiles at various angular positions ψ (its definition can be found in Fig. 33(a)). As seen, the numerical solutions compare well with the experimental data of Kuehn et al. [67] at various angular positions. While it is clear that the accuracy of the numerical result can be enhanced via pressure mesh refinement as reported in Fig. 35, it is worth to mention here that the solutions of the case employing $N_m = 7536$ ($l_{avg} = 1$ mm) are already grid-independent.

5. Conclusion

In the newly proposed Unstructured Moving Particle Pressure Mesh (UMPPM) method presented in the current work, a hybrid unstructured pressure mesh coupled with the moving particle strategy proposed in the original Moving Particle Semi-implicit (MPS) method is used to solve the incompressible flow problems involving complex boundary shapes. In order to address the inconsistency issue of the original Moving Particle Pressure Mesh (MPPM) method, the viscous term has been discretized by a numerically consistent Laplacian model (denoted as the Consistent Particle Method (CPM) in the open literature). The CPM formulation used to discretize the viscous term has been solved in its implicit form in order to get rid of the constraint of time step size in the original explicit scheme. The numerical solutions of UMPPM have been compared with the benchmark solutions for several incompressible isothermal and non-isothermal flow problems involving geometrically complex flow boundaries. Good agreement has been found between the UMPPM and benchmark solutions.

Currently, we are looking for ways to address the free-surface flow problem by UMPPM via the single fluid formulation, by enforcing the vanishing shear stress boundary condition on the free surface. The advantage of using massless particles to represent the free surface (as considered in the PIC method), however, is simply lost in UMPPM due to the incorporation of particle deletion/addition procedure outlined above. Also, the argument of violation of momentum conservation at the particle level due to the particle deletion/addition procedure should deserve further attention in the future work.

Acknowledgments

The financial supports provided by the Ministry of Education Malaysia (Fundamental Research Grant Scheme Ref. No: FRGS/2/2013/TK01/UNITEN/02/1) and the Ministry of Science, Technology and Innovation (MOSTI) Malaysia (Project No: 06-02-03-SF0258) are greatly acknowledged.

References

- [1] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, *Nucl. Sci. Eng.* 123 (1996) 421–434.
- [2] Y. Nakanishi, T. Fujii, S. Kawaguchi, Numerical and experimental investigations of the flow in a stationary Pelton bucket, *J. Fluid Sci. Technol.* 4 (3) (2009) 490–499.
- [3] Z.G. Sun, Y.Y. Liang, G. Xi, Numerical simulation of the flow in straight blade agitator with the MPS method, *Internat. J. Numer. Methods Fluids* 67 (2011) 1960–1972.
- [4] R. Chen, W. Tian, G.H. Su, S. Qiu, Y. Ishiwatari, Y. Oka, Numerical investigation on bubble dynamics during flow boiling using moving particle semi-implicit method, *Nucl. Eng. Des.* 240 (2010) 3830–3840.
- [5] S. Yashiro, T. Okabe, K. Matsushima, A numerical approach for injection molding of short-fiber-reinforced plastics using a particle method, *Adv. Compos. Mater.* 20 (2011) 503–517.
- [6] T. Kawahara, Y. Oka, Ex-vessel molten core solidification behavior by moving particle semi-implicit method, *J. Nucl. Sci. Technol.* 49 (12) (2012) 1156–1164.
- [7] T. Kon, S. Natsu, S. Ueda, R. Inoue, T. Ariyama, Influence of physical properties of melt on liquid dripping in packed bed analyzed by MPS Method, *ISIJ Int.* 53 (4) (2013) 590–597.
- [8] K.C. Ng, E.Y.K. Ng, W.H. Lam, Lagrangian simulation of steady and unsteady laminar mixing by plate impeller in a cylindrical vessel, *Ind. Eng. Chem. Res.* 52 (29) (2013) 10004–10014.
- [9] K.C. Ng, E.Y.K. Ng, Laminar mixing performances of baffling, shaft eccentricity and unsteady mixing in a cylindrical vessel, *Chem. Eng. Sci.* 104 (2013) 960–974.
- [10] K. Shibata, S. Koshizuka, Numerical analysis of shipping water impact on a deck using a particle method, *Ocean Eng.* 34 (2007) 585–593.
- [11] K. Kakuda, Y. Ushiyama, S. Obara, J. Toyotani, S. Matsuda, H. Tanaka, K. Katagiri, Flow simulations in a liquid ring pump using a particle method, *Comput. Model. Eng. Sci.* 66 (3) (2010) 215–226.
- [12] N. Hirata, K. Anzai, Numerical simulation of shrinkage formation of pure Sn casting using particle method, *Mater. Trans.* 52 (10) (2011) 1931–1938.
- [13] T. Kon, S. Natsui, S. Ueda, R. Inoue, T. Ariyama, Numerical simulation of dripping behavior of droplet in packed bed using particle method, *ISIJ Int.* 52 (9) (2012) 1565–1573.

- [14] T. Okabe, H. Matsutani, T. Honda, S. Yashiro, Numerical simulation of microscopic flow in a fiber bundle using the moving particle semi-implicit method, *Composites A* 43 (2012) 1765–1774.
- [15] S. Koshizuka, A. Nobe, Y. Oka, Numerical analysis of breaking waves using the moving particle semi-implicit method, *Internat. J. Numer. Methods Fluids* 26 (1998) 751–769.
- [16] S. Koshizuka, Y. Oka, Moving particle semi implicit method: Fully Lagrangian analysis of incompressible flows, in: *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2000*, Barcelona, 11–14 September 2000, 2000, pp. 1–16.
- [17] J.S. Kouh, Simulation of a ship with partially filled tanks rolling in waves by applying moving particle semi-implicit method, in: *International Conference on Engineering Education ICEE2007*, Coimbra, Portugal, September 3–7 2007, 2007, p. 6.
- [18] A. Shakibaeinia, Y.C. Jin, Lagrangian multiphase modeling of sand discharge into still water, *Adv. Water Resour.* 48 (2012) 55–67.
- [19] Z.G. Sun, G. Xi, X. Chen, A numerical study of stir mixing of liquids with particle method, *Chem. Eng. Sci.* 64 (2009) 341–350.
- [20] M. Sakai, Y. Shigeto, X. Sun, T. Aoki, T. Saito, J. Xiong, S. Koshizuka, Lagrangian–Lagrangian modeling for a solid–liquid flow in a cylindrical tank, *Chem. Eng. J.* 200–202 (2012) 663–672.
- [21] D. Li, Z.G. Sun, X. Chen, G. Xi, L. Liu, Analysis of wall boundary in moving particle semi-implicit method and a novel model of fluid–wall interaction, *Int. J. Comput. Fluid Dyn.* (2015) <http://dx.doi.org/10.1080/10618562.2015.1028924>.
- [22] H. Xie, S. Koshizuka, Y. Oka, Modelling of a single drop impact onto liquid film using particle method, *Internat. J. Numer. Methods Fluids* 45 (2004) 1009–1023.
- [23] 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.
- [24] B.H. Lee, J.C. Park, M.H. Kim, S.C. Hwang, Step-by-step improvement of MPS method in simulating violent free-surface motions and impact loads, *Comput. Methods Appl. Mech. Engrg.* 200 (2011) 1113–1125.
- [25] H. Akimoto, Numerical simulation of the flow around a planing body by MPS method, *Ocean Eng.* 64 (2013) 72–79.
- [26] Y.H. Hwang, A moving particle method with embedded pressure mesh (MPPM) for incompressible flow calculations, *Numer. Heat Transfer B* 60 (2011) 370–398.
- [27] K.C. Ng, Y.H. Hwang, T.W.H. Sheu, On the accuracy assessment of Laplacian models in MPS, *Comput. Phys. Comm.* 185 (10) (2014) 2412–2426.
- [28] 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.
- [29] F.H. Harlow, A machine calculation method for hydrodynamic problems, in: *Technical Report LAMS-1956*, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1955.
- [30] D.M. Kelly, Q. Chen, J. Zang, PICIN: A particle-in-cell solver for incompressible free surface flows with two-way fluid–solid coupling, *SIAM J. Sci. Comput.* 37 (3) (2015) B404–B424.
- [31] T. Matsunaga, K. Shibata, K. Murotani, S. Koshizuka, Hybrid grid-particle method for fluid mixing simulation, *Comput. Part. Mech.* 2 (2015) 233–246.
- [32] J. Liu, S. Koshizuka, Y. Oka, A hybrid particle-mesh method for viscous, incompressible, multiphase flows, *J. Comput. Phys.* 202 (2005) 65–93.
- [33] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209–240.
- [34] K.C. Ng, A collocated finite volume embedding method for simulation of flow past stationary and moving body, *Comput. & Fluids* 38 (2) (2009) 347–357.
- [35] 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.
- [36] M. Tanaka, T. Masunaga, Stabilization and smoothing of pressure in MPS method by quasi-compressibility, *J. Comput. Phys.* 229 (2010) 4279–4290.
- [37] M. Kondo, S. Koshizuka, Improvement of stability in moving particle semi-implicit method, *Internat. J. Numer. Methods Fluids* 65 (2011) 638–654.
- [38] S. Natsui, H. Takai, T. Kumagai, T. Kikuchi, R.O. Suzuki, Stable mesh-free moving particle semi-implicit method for direct analysis of gas–liquid two-phase flow, *Chem. Eng. Sci.* 111 (2014) 286–298.
- [39] J. Stam, Stable fluids, in: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'99*, Los Angeles, August 8–13 1999, 1999, pp. 121–128.
- [40] M. Jin, Q. Chen, Accelerating fast fluid dynamics with a coarse-grid projection scheme, *HVAC R Res.* 20 (8) (2014) 932–943.
- [41] M. Jin, W. Zuo, Q. Chen, Improvements of fast fluid dynamics for simulating airflow in buildings, *Numer. Heat Transfer B* 62 (6) (2012) 419–438.
- [42] Y. Fukuzawa, H. Tomiyama, K. Shibata, S. Koshizuka, Numerical analysis of high viscous non-Newtonian fluid flow using the MPS method, *Transactions of JSCEs*, Paper No 2014007, 2014.
- [43] A. Souto-Iglesias, F. Macia, L.M. Gonzalez, J.L. Cercos-Pita, On the consistency of MPS, *Comput. Phys. Comm.* 184 (3) (2013) 732–745.
- [44] C.G. Koh, M. Luo, M. Gao, W. Bai, Modelling of liquid sloshing with constrained floating baffle, *Comput. Struct.* 122 (2013) 270–279.
- [45] M. Gao, Numerical simulation of liquid sloshing in rectangular tanks using consistent particle method and experimental verification (Ph.D. thesis), National University of Singapore, Singapore, 2011.
- [46] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows (Ph.D. thesis), Imperial College London (University of London), UK, 1996.
- [47] B. Niceno, A three dimensional, finite volume method for incompressible Navier Stokes equations on unstructured, staggered grids, in: *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, September 5–8, 2006, Egmond aan Zee, The Netherlands, paper 196, 2006.
- [48] K.C. Ng, M.Z. Yusoff, E.Y.K. Ng, Multigrid solution of Euler equations using high-resolution NVD differencing scheme for unstructured meshes, *Prog. Comput. Fluid Dyn.* 6 (7) (2006) 389–401.

- [49] K.C. Ng, M.Z. Yusoff, E.Y.K. Ng, Parametric study of an improved GAMMA differencing scheme based on normalized-variable formulation for low-speed flow with artificial compressibility technique, *Numer. Heat Transfer B* 50 (6) (2006) 561–584.
- [50] K.C. Ng, M.Z. Yusoff, E.Y.K. Ng, Higher-order bounded differencing schemes for compressible and incompressible flows, *Internat. J. Numer. Methods Fluids* 53 (1) (2007) 57–80.
- [51] K.C. Ng, E.Y.K. Ng, M.Z. Yusoff, T.K. Lim, Applications of high-resolution schemes based on normalized variable formulation for 3D indoor airflow simulations, *Internat. J. Numer. Methods Engrg.* 73 (7) (2008) 948–981.
- [52] H.Y. Yoon, S. Koshizuka, Y. Oka, A particle-gridless hybrid method for incompressible flows, *Internat. J. Numer. Methods Fluids* 30 (1999) 407–424.
- [53] A. Shakibaeinia, Y.C. Jin, MPS mesh-free particle method for multiphase flows, *Comput. Methods Appl. Mech. Engrg.* 229–232 (2012) 13–26.
- [54] M. Tabbara, T. Blacker, T. Belytschko, Finite element derivative recovery by moving least square interpolants, *Comput. Methods Appl. Mech. Engrg.* 117 (1994) 211–223.
- [55] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, *J. Comput. Phys.* 191 (2) (2003) 448–475.
- [56] H.A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 13 (2) (1992) 631–644.
- [57] A. Khayyer, H. Gotoh, Development of CMPS method for accurate water-surface tracking in breaking waves, *Coastal Eng. J.* 50 (2) (2008) 179–207.
- [58] M. Ellero, M. Serrano, P. Espanol, Incompressible smoothed particle hydrodynamics, *J. Comput. Phys.* 226 (2007) 1731–1752.
- [59] S. Adami, X.Y. Hu, N.A. Adams, A transport-velocity formulation for smoothed particle hydrodynamics, *J. Comput. Phys.* 241 (2013) 292–307.
- [60] 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.
- [61] E. Erturk, B. Dursun, Numerical solutions of 2D steady incompressible flow in a driven skewed cavity, *ZAMM J. Appl. Math. Mech.* 87 (2007) 377–392.
- [62] 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.
- [63] C. Shu, H. Xue, Y.D. Zhu, Numerical study of natural convection in an eccentric annulus between a square outer cylinder and a circular inner cylinder using DQ method, *Int. J. Heat Mass Transfer* 44 (2001) 3321–3333.
- [64] Y. Peng, Y.T. Chew, C. Shu, Numerical simulation of natural convection in a concentric annulus between a square outer cylinder and a circular inner cylinder using the Taylor-series-expansion and least-squares-based lattice Boltzmann method, *Phys. Rev. E* 67 (2003) 026701.
- [65] F. Moukalled, S. Acharya, Natural convection in the annulus between concentric horizontal circular and square cylinders, *J. Thermophys Heat Transfer* 10 (3) (1996) 524–531.
- [66] Y. Liang, Z. Sun, G. Xi, L. Liu, Numerical models for heat conduction and natural convection with symmetry boundary condition based on particle method, *Int. J. Heat Mass Transfer* 88 (2015) 433–444.
- [67] T.H. Kuehn, R.J. Goldstein, An experimental and theoretical study of natural convection in the annulus between horizontal concentric cylinders, *J. Fluid Mech.* 74 (4) (1976) 695–719.