Material point method and smoothed particle hydrodynamics simulations of fluid flow problems: a comparative study

Zheng Sun
Key Laboratory of Soft Machines and Smart Devices of Zhejiang Province, Department of Engineering Mechanics, Zhejiang University, Hangzhou, Zhejiang 310027, China
Email: sunzheng@zju.edu.cn

Haiqiao Li
Laboratory of Energy and Environment and Computational Fluid Dynamics, North University of China, Taiyuan, Shanxi 030051, China
Email: lihaiqiao8@sina.com

Yong Gan*
Key Laboratory of Soft Machines and Smart Devices of Zhejiang Province, Department of Engineering Mechanics, Zhejiang University, Hangzhou, Zhejiang 310027, China
Email: ganyong@zju.edu.cn
*Corresponding author

Hantao Liu
Laboratory of Energy and Environment and Computational Fluid Dynamics, North University of China, Taiyuan, Shanxi 030051, China
Email: lht@nuc.edu.cn

Zhilong Huang
Key Laboratory of Soft Machines and Smart Devices of Zhejiang Province, Department of Engineering Mechanics, Zhejiang University, Hangzhou, Zhejiang 310027, China
Email: zlhuang@zju.edu.cn

Lisha He
School of Civil Engineering and Architecture, Nanchang University, Nanchang, Jiangxi 330031, China
Email: lisha.he@outlook.com

Abstract: The material point method (MPM) and the smoothed particle hydrodynamics (SPH) are two commonly used particle-based methods for solving large-deformation problems. Especially, the SPH has been widely applied to fluid dynamics problems, while the MPM performance in fluid dynamics simulations has rarely been investigated. In this study, the
capabilities of the MPM and the SPH in simulating fluid dynamics problems have been quantitatively examined and compared through three example problems, i.e., Poiseuille and Couette flows and water dam break flows. Both numerical methods could yield the results in good agreements with the theoretical and experimental results. Without requiring neighbour search and additional boundary particles, the MPM exhibits significantly higher computational efficiency as compared with the SPH. The comparisons also demonstrate that the MPM has higher accuracy and faster convergence than the SPH. It is shown that the MPM could be a promising alternative to the SPH for the fluid dynamics simulations. Future work for the improvement of the MPM in fluid dynamics modelling is discussed.

Keywords: material point method; MPM; smoothed particle hydrodynamics; SPH; fluid flow problems; particle methods; computational fluid dynamics; Poiseuille flow; Couette flow; dam break flow.


Biographical notes: Zheng Sun received his BE in Engineering Mechanics in 2009 and MS in Solid Mechanics in 2012 from the Shandong University, respectively. He is currently doing his research in computational mechanics at the Department of Engineering Mechanics at Zhejiang University as a PhD student.

Haiqiao Li received his BSc in Thermal Energy and Power Engineering and MSc in Power Engineering and Engineering Thermophysics from the North University of China in 2007 and 2011, respectively. Currently, he is a PhD student at Laboratory of Energy and Environment and Computational Fluid Dynamics of North University of China in Chair of Computational Fluid Dynamics with meshfree method.

Yong Gan received his PhD in 2008 from the University of Missouri-Columbia. He is currently an Associate Professor at the Zhejiang University focusing on the computational mechanics and nanoscience.

Hantao Liu received his BSc in Thermal Engineering from the Nanjing University of Science and Technology in 1999. He received his MSc in Thermal Engineering from the Shandong University in 2005. He received his PhD in Armament Science and Technology from the North University of China in 2012. Currently, he is an Associate Professor at the Laboratory of Energy and Environment and Computational Fluid Dynamics of North University of China. He has been a Visiting Scientist at the University of Missouri. His interests are mainly focused on the development of meshfree methods for multiphase flow and fluid-structure interaction.

Zhilong Huang is a Professor of Mechanics at the Zhejiang University. He is currently doing his research in the fields of nonlinear stochastic dynamics, complex network dynamics and analysis and calculation of complex structures.

Lisha He received her Diploma in Engineering, including BEng and MSc, in 2011 from the Huazhong University of Science and Technology and Durham University. She received her PhD in Computational Mechanics from the Durham University in 2016. Currently, she is a Lecturer at the School of Civil Engineering and Architecture in Nanchang University, China. Her research interests focus on applying and improving mesh-free methods to solve large deformation, impact and shock wave problems.

1 Introduction

Fluid flow is a common physical phenomenon and of great importance and interest in many industrial fields such as civil, mechanical, chemical, aerospace and petroleum engineering (Chwang and Wu, 1975; Purcell, 1977). Due to their strong complexity and nonlinearity, the fluid flow problems are usually studied by means of numerical methods. In general, the motion of materials can be described in an Eulerian or Lagrangian way. As can be found in open literature (Chen et al., 2016; Colella, 1985; Desjardins et al., 2008; Harten et al., 1987; Jiang and Shu, 1996), Eulerian methods have been widely used to simulate fluid flows, for one major advantage of avoiding mesh entanglement associated with large deformations that confronts the mesh-based Lagrangian approaches, e.g., finite element method. However, Eulerian methods suffer from the limitation in tracking material interfaces. Moreover, modelling of fluid flows with free surface is also a challenging task for Eulerian approaches. Over the past several decades, meshfree methods have been increasingly become an important tool for solving large-deformation
problems, where continuum bodies are represented by a set of Lagrangian particles and the deformation of bodies is described by the movements of particles. Among them are two typical examples, namely smoothed particle hydrodynamics (SPH) and material point method (MPM).

As one of the earliest mesh free methods, SPH was initially proposed by Lucy (1977), and Gingold and Monaghan (1977) for astrophysical problems. The extension of the SPH method in modelling fluid flow problems were first developed by Monaghan (1994) and Morris et al. (1997). Henceforth, a large number of efforts have been made to propose different approaches to improving the accuracy of SPH for fluid dynamics. For instance, Sigalotti et al. (2003) presented one SPH scheme for solving the motion equations of a viscous fluid flow. Oger et al. (2007) applied the renormalisation scheme in the SPH for enhanced accuracy. Hosseini and Feng (2011) demonstrated that a rotational pressure-correction scheme for pressure boundary conditions could result in accurate pressure and velocity near open boundaries and solid objects. Adami et al. (2012) used a generalised wall boundary condition scheme in the SPH method to handle the problem domain with sharp corners and complex geometries. Barcarolo et al. (2014) carried out an adaptive particle refinement procedure with time-varying particle distribution for higher resolutions.

The MPM was developed by extending a hydrodynamics code FLIP (Brackbill and Ruppel, 1986) to solid mechanics problems. So far, the MPM has been widely used in solid mechanics simulations, particularly for problems involving large deformations (Aimene and Nairn, 2015; Lian et al., 2012, 2015; Sadeghirad et al., 2011, contact (Huang et al., 2011; Ma et al., 2014), high-velocity impact process (Liu et al., 2013, 2015) and blast fragmentation (Hu and Chen, 2006), fracture and crack propagation (Gao and Nairn, 2004, 2006; Nairn, 2003), granular material (Bardenhagen et al., 2000; Wieckowski et al., 1999), etc. Although the MPM was originally designed to solve solid mechanics problems, it can also handle fluid dynamics problems in the same manner. Recently, there is a growing interest in the MPM modelling of fluid dynamics. For example, fluid-membrane interaction problems involving fluid dynamics were simulated by York et al. (1999, 2000), Gan et al. (2011) and Lian et al. (2014). Sulsky et al. (2007) modelled sea ice dynamics problems using the MPM. Chen et al. (2012) studied the fluid-solid interactions induced by surface tension using generalised interpolation material point method (GIMP). The works on the MPM that consider the coupling behaviour of solid skeleton and pore fluid based on the mixture theory have been conducted by many researchers (Zhang et al., 2009; Zheng et al., 2013; Bandara et al., 2016).

Due to the use of a background Eulerian mesh for solving the momentum equation, the MPM boundary conditions can be applied to the corresponding background grid nodes as easily as the FEM (Ma et al., 2009). In the MPM computation, the state information carried by the material points is mapped to the grid nodes, and no neighbour particle searching is required. In comparison, additional particles and special treatments are inherently needed in the SPH for applying boundary conditions, and moreover, the search for neighbouring particles is conducted at each SPH time step. Therefore, the MPM is expected to be more computationally efficient than the SPH, and to be advantageous over the SPH in fluid dynamics simulations. This implies that a systematic work is necessary for the quantitative comparisons on the efficiency and accuracy of the SPH and the MPM in fluid dynamics simulation, which, however, has not yet been conducted to our knowledge.

To provide a straight comparison of the capabilities of the MPM and the SPH in modelling fluid dynamics phenomena, three fluid flow problems, namely, Poiseuille and Couette flows and water dam break flows, have been simulated using the MPM and the SPH in this work. The remaining sections of this paper are organised as follows. The governing equations of the MPM and the SPH are briefly presented in Section 2. In Section 3, the MPM and SPH simulations of three fluid flow problems are described and discussed, including the comparisons of the numerical solutions with the theoretical and experimental results. Finally, concluding remarks are drawn in Section 4.

2 Governing equations and numerical implementation of MPM and SPH

The procedures for deriving the governing equations for the MPM and SPH have been well discussed elsewhere (Gingold and Monaghan, 1977; Liu and Liu, 2003; Lucy, 1977; Sulsky et al., 1994), and are briefly presented here for completeness. The continuum conservation equations of mass and momentum are expressed, respectively

\[
\frac{dp}{dt} + \rho \nabla \cdot \mathbf{v} = 0 \tag{1}
\]

\[
\rho \ddot{\mathbf{u}} = \nabla \cdot \mathbf{\sigma} + \rho \mathbf{b} \tag{2}
\]

where \( \rho \) is density, \( \mathbf{v} \) is the velocity, \( \mathbf{a} \) is the acceleration, \( \mathbf{\sigma} \) is the Cauchy stress, and \( \mathbf{b} \) is the specific body force. The energy equation is not considered due to the negligible thermal effect in this work.

Equations (1) and (2) can be solved numerically in the solving domain \( \Omega \) with initial conditions and boundary conditions.

- **Initial conditions:**
  \[
  \rho(x, 0) = \rho(x), \quad x \in \Omega \tag{3}
  \]
  \[
  \mathbf{v}(x, 0) = \mathbf{v}(x), \quad x \in \Omega \tag{4}
  \]
  \[
  \mathbf{\sigma}(x, 0) = \mathbf{\sigma}(x), \quad x \in \Omega \tag{5}
  \]

- **Boundary conditions:**
  \[
  \mathbf{u}(x, t) = \mathbf{u}(t), \quad x \in \partial \Omega_u \tag{6}
  \]
  \[
  \mathbf{\sigma}(x, t) \cdot \mathbf{n}_b = \mathbf{t}_b(t), \quad x \in \partial \Omega_r \tag{7}
  \]
where $\Omega$ is the current configuration of the continuum body; $\rho(x)$, $v(x)$, and $\mathbf{\sigma}(x)$ denote the initial mass density, velocity and initial stress of the continuum body, respectively; $\partial \Omega$ and $\partial \Omega_s$ are the parts of the boundary subject to the prescribed displacement and traction, respectively; $u(t)$ and $\mathbf{t}_b(t)$ represent the prescribed displacement and traction, respectively. $n_b$ is the outer norm vector of the traction boundary.

### 2.1 MPM formulation

In the MPM, a continuum body is represented by a finite number of discrete Lagrangian particles (also called material points). The deformations of continuum body are described by tracking the movements of material points. Since the mass of each material point remains unchanged over time, the conservation equation of mass equation (1) is automatically satisfied. The discretised MPM equation of momentum becomes

$$\int_{\Omega} \rho \mathbf{w} \cdot a d\Omega = -\int_{\Omega} \rho \mathbf{\sigma} \cdot \nabla \mathbf{w} d\Omega + \int_{\partial \Omega_b} \mathbf{t}_b \cdot \mathbf{w} dS \quad (8)$$

where $\rho \mathbf{\sigma} = \mathbf{\sigma}$, and $\mathbf{w}$ denotes the test function.

At time level $k$, each material point carries the material properties and state variables, including mass $M_p$, density $\rho_p^k$, velocity $v_p^k$, stress $\mathbf{\sigma}_p^k$, strain $\mathbf{\varepsilon}_p^k$, etc. Thus, the integrals in equation (8) are calculated by the sums of quantities at material points,

$$\sum_{p=1}^{N_p} M_p \mathbf{w}[X_p(t), t] \cdot a[X_p(t), t] =$$

$$-\sum_{p=1}^{N_p} M_p \mathbf{\sigma}_p^k \cdot \frac{\partial}{\partial t} \mathbf{w}[X_p(t), t] + \int_{\partial \Omega_b} \mathbf{t}_b \cdot \mathbf{w} dS \quad (9)$$

$$+ \sum_{p=1}^{N_p} M_p \mathbf{w}[X_p(t), t] \cdot b[X_p(t), t]$$

where $X_p(t)$ is the position of material point $p$ at time $t$. Meanwhile, an Eulerian background mesh as shown in Figure 1 is defined to calculate the gradient terms and solve the momentum equation.

By using the conventional finite element shape function, the filed variables and test function at the material points are interpolated by the corresponding values at the background mesh nodes. Then, with the use of lumped mass, the discretised MPM equation of momentum becomes

$$m_p^k \mathbf{a}_p^k = \mathbf{f}_{\text{int},k} + \mathbf{f}_{\text{ext},k} \quad (10)$$

$$\mathbf{f}_{\text{int},k} = -\sum_{p=1}^{N_p} M_p \mathbf{G}_p^k \cdot \mathbf{\sigma}_p^k \quad (11)$$

$$\mathbf{f}_{\text{ext},k} = \mathbf{b}_p^k + \mathbf{t}_b^k \quad (12)$$

$$\mathbf{G}_p^k = \nabla N_i(X_p^k) \quad (13)$$

where $m_p^k = \sum_{p=1}^{N_p} M_p N_i(X_p^k)$ denotes the lumped mass at the $i$th grid node at time level $k$, $\mathbf{f}_{\text{int},k}$ and $\mathbf{f}_{\text{ext},k}$ represent the $i$th grid node internal and external force tensor at time level $k$, respectively; $N_i(X_p^k)$ is the shape function of node $i$ associated with spatial material points $X_p^k$ at time level $k$.

In the present study, an explicit Euler algorithm is used to solve equation (10) for nodal accelerations. The grid node velocities at time level $k + 1$, are obtained by

$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + \mathbf{a}_p^k \Delta t \quad (14)$$

Then, the positions and velocities of the material point $p$ at time level $k + 1$, $\mathbf{X}_p^{k+1}$ and $\mathbf{v}_p^{k+1}$, can be calculated from the interpolations of grid node velocities and accelerations, respectively

$$\mathbf{X}_p^{k+1} = \mathbf{X}_p^k + \sum_{i=1}^{N_p} \mathbf{v}_i^k N_i(X_p^k) \Delta t \quad (15)$$

$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + \sum_{i=1}^{N_p} \mathbf{a}_i^k N_i(X_p^k) \Delta t \quad (16)$$

where $\Delta t$ is the time step for the explicit integration. The strain increments are updated by mapping the updated particle moment back to the grid nodes

$$\Delta \mathbf{\varepsilon}_p^{k+1} = \frac{\Delta t}{2} \left[ \mathbf{G}_p^k \mathbf{v}_p^{k+1} + \left( \mathbf{G}_p^k \mathbf{v}_p^{k+1} \right)^T \right] \quad (18)$$

The stress increment can be derived from the given strain increment based on the constitutive model, and then used to update the particle stress tensor,

$$\mathbf{\sigma}_p^{k+1} = \mathbf{\sigma}_p^k + f(\Delta \mathbf{\varepsilon}_p^{k+1}) \quad (19)$$

where $f$ describes the constitutive law of the material.
2.2 SPH formulation

Similar to the MPM, the problem domain is also discretised by a finite set of particles carrying the individual mass \( m_i \), velocity \( v_i \), and other properties at position \( r_i \) in the SPH. Kernel approximations and particle approximations are two main steps for obtaining an SPH formulation. In kernel approximations, the integral form of smoothing functions and its derivatives are used to represent the variable field of the continuous domain. In particle approximations, a set of particles is used to represent the variable field of the continuous domain. In particle approximations, a set of particles are used to discretise the problem domain. For example, a function \( A(r) \) and its derivative \( \nabla A(r) \) can be written in the SPH forms as follow:

\[
A(r) = \int A(r') W(r - r', h) \, dr' = \sum_{j=1}^{N} \frac{m_j A(r_j)}{p_j} W(r - r_j, h) 
\]

\[
\nabla A(r) \approx \int \nabla A(r') W(r - r', h) \, dr' \approx \sum_{j=1}^{N} \frac{m_j A(r_j)}{p_j} \nabla W(r - r_j, h) 
\]

where the subscript \( j \) denotes the neighbouring particles in the support domain, \( W \) is the smoothing function defined over the support domain, \( r \) is the position vector of the particle, \( h \) is the smooth length, \( m_j \) and \( p_j \) denote the mass and the density of the particle \( j \) in support domain, respectively.

The kernel function plays an important role in SPH simulation (Shao et al., 2012a). In this paper, we chose the cubic spline kernel function as it has continuous second-order derivative (Monaghan, 1985)

\[
W_y = \frac{10}{7\pi h_y^3} \left\{ \begin{array}{ll}
1 - \frac{3}{2} q_y^2 + \frac{3}{4} q_y^3, & 0 \leq q_y < 1 \\
\frac{1}{4} (2 - q_y)^2, & 1 \leq q_y \leq 2 \\
0, & q_y > 2
\end{array} \right.
\]

where \( q_y = \|r_i - r_j\| / h_y \) is the relative distance between particles \( j \) and \( i \), with \( h_y \) being the smoothing length that defines the size of the support domain for the smoothing function. The choice of \( h_y \) should be careful because it directly affects the simulation efficiency and accuracy.

Here, we take \( h_y \) equal to 1.1–1.2 times of the average particle space (Monaghan, 1994).

Based on the SPH discretisation algorithm presented in equations (20) and (21), the continuity equation (1) and the momentum equation (2) without artificial viscosity term can be expressed as the following SPH form (Liu and Liu, 2003):

\[
\rho_i = -\sum_{j=1}^{N} \frac{m_j (v_i - v_j) \cdot \nabla W(r_i - r_j, h_i)}{\rho_i} (23)
\]

\[
a_i = -\sum_{j=1}^{N} \frac{m_j (P_{ij} - P_j)}{\rho_i \rho_j} \nabla W(r_i - r_j, h_i) + \Theta + g_i (24)
\]

where the subscripts \( i \) and \( j \) denote the considered and the neighbouring particles in the support domain, and \( \Theta \) and \( g \) denote the diffusion term and the gravity acceleration of the particle which is one form of the body force in equation (2), respectively. Considering a laminar flow, the diffusion term in momentum equation (24) has the form of:

\[
\Theta = v_0 \nabla^2 \nu (25)
\]

where \( v_0 \) is the kinematic viscosity of fluid. The laminar stress term is simplified (Lo and Shao, 2002) as:

\[
\left( v_0 \nabla^2 \nu \right) = \sum_{j=1}^{N} m_j \left[ \frac{4\nu |r_i - r_j|}{(\rho_i + \rho_j)|r_i|^2 + \eta^2} \right] v_j (26)
\]

in which \( r_y = r_i - r_j, \nu_y = v_i - v_j \), and \( \eta \) is used to ensure a non-zero denominator and \( \eta = 0.1 \, h \).

In fluid dynamic simulations, the artificial viscosity is usually considered as the diffusion term to alleviate pressure oscillations caused by particles’ unphysical concentration, which takes the form of

\[
\Pi_y = \left\{ \begin{array}{ll}
-\alpha \frac{\nabla \cdot \nu_y}{\nu_y}, & v_y \nu_y < 0 \\
0, & v_y \nu_y \geq 0
\end{array} \right.
\]

(27)

where \( \mu_y = (h \nu_y \cdot r_y) / (|r_y|^2 + \eta^2), \eta^2 = 0.01 \, h^2, \, \nu_y \) is the average value of the numerical sound velocities of particles \( i \) and \( j \), and \( \alpha \) is a parameter with its value depending on the characteristics of specific problem. The selection of \( \alpha \) is important because small \( \alpha \) cannot effectively smooth the
oscillations in dynamic problems, but large alpha may cause high numerical dissipations and lead to large errors. We take $\alpha = 0.01$ in all the simulations in this paper (Monaghan, 1994). Thus, combining equations (24), (25), (26) and (27), the momentum equation could be expressed as:

$$a_i = -\sum_{j=1}^{N} m_j \left( \frac{p_i}{\rho^2_i} - \frac{p_j}{\rho^2_j} + \Pi_{ij} \right) \nabla_i \delta_i \left( \mathbf{r}_i - \mathbf{r}_j, h_0 \right) + \sum_{j=1}^{N} m_j \left[ \frac{4\eta_i}{\rho_i + \rho_j} \frac{\delta_i \left( \mathbf{r}_i - \mathbf{r}_j, h_0 \right)}{\rho^2_i + \eta^2_i} \right] \mathbf{v}_i + \mathbf{g}_i.$$

(28)

It is known that the standard SPH method suffers from low computational accuracy as the computation proceeds and the particles distributions become irregular. Especially the continuity equations are no longer satisfied near the boundary (Adami et al., 2012). Therefore, the periodic density and kernel function corrections are necessary for solving this problem. The density correction and regularised kernel function correction are applied every 30 time steps and can be described as follows (Shao et al., 2012a):

$$\rho_i^{\text{new}} = \sum_{j=1}^{N} \frac{\rho_j W_{ij}^{\text{new}} m_j}{\rho_j} = \sum_{j=1}^{N} m_j W_{ij}^{\text{new}}$$

(29)

$$W_{ij}^{\text{new}} = \frac{W_{ij}}{\sum_{j=1}^{N} \frac{m_j}{\rho_j}}$$

(30)

2.3 Boundary conditions

In the MPM, due to the use of shape functions and background mesh, the essential boundary conditions can be implemented at the background grid nodes. Furthermore, because of the single-valued velocity field of grid nodes in the MPM, the velocity boundary condition can also be easily applied at the background grid nodes, and the no-slip contact conditions are automatically handled without requiring additional treatments.

In comparison, the implementation of the boundary conditions is one of the major challenges for the SPH. In this work, a coupled dynamic boundary condition is applied by using the repulsive and ghost particles as shown in Figure 2. More details for this boundary treatment have been given by Shao et al. (2012b).
2.4 Time integration
The explicit forward Euler time integration and the explicit predictor-corrector time discretisation are used in the MPM and SPH scheme, respectively, for a balance between the computational accuracy and efficiency. The critical time step is obtained from the Courant-Friedrichs-Lewy (CFL) condition for numerical stability in both MPM and SPH simulations.

\[ \Delta t_{cr} = \frac{d_c}{c} \quad (31) \]

where \( \Delta t_{cr} \) is the critical time step, \( d_c \) denotes the characteristic size, and \( c \) is the sound speed of fluid. For the MPM and SPH, \( d_c \) is equal to the smallest background mesh size and the smallest smoothing length, respectively.

2.5 Nearest neighbouring particle searching in the SPH
In the SPH method, all-pair search algorithm, tree search algorithm and linked-list search algorithm are commonly used to search for the nearest neighbouring particles. The all-pair search approach identifies the neighbouring particles for each particle by looping over all the particles, and its computational complexity is of \( O(N^2) \) for a system with \( N \) particles. Therefore, it is used only for problems with very small scale. Tree search algorithm presents a good performance in the neighbouring particle search for a large system with variable smoothing lengths. The complexity of the tree search algorithm is of order \( O(N \log N) \), much less than the all-pair search method. In this paper, the linked-list search algorithm is used for its high efficiency in the simulations with a constant smoothing length. In the linked-list search algorithm for two-dimensional problems, the computational domain is divided into square cells of side \( \kappa h \) (where \( \kappa = 2 \)), as shown in Figure 3. For the particles inside cell \((i, j)\), only the interactions with particles of five neighbouring cells of \((i, j), (i + 1, j), (i - 1, j + 1), (i, j + 1), (i + 1, j + 1)\) need to be considered. If the average number of particles per cell is sufficiently small, the complexity of the linked-list algorithm is of order \( O(N) \).

3 Numerical examples
We consider the incompressible Newtonian fluids in all example problems. The stress-strain relation for Newtonian fluids is given as

\[ \sigma_p = 2 \mu \dot{e}_p - \frac{2}{3} \mu tr(\dot{e}_p) \delta - P_p \delta \quad (32) \]

where \( \mu \) is dynamics viscosity, \( P_p \) is pressure of fluid particle \( p \) that can be determined using an equation of state (EOS), and \( \delta \) denotes Kronecker delta. To accelerate the computation, a quasi-incompressible EOS for fluids as suggested by Monaghan (1994) is used in both MPM and SPH simulations

\[ P = K \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \quad (33) \]

where \( \gamma = 7.0 \) and \( K \) is the artificial bulk modulus. The serial MPM and SPH codes are compiled using Intel Fortran 90 Compiler on a 64-bit Linux workstation with 64 GB memory, and all simulations are performed using single AMD Opteron 6272 CPU core. The computing time is counted by the intrinsic Fortran procedure of CPU_TIME. The memory storages used for both methods are monitored by the Linux command of TOP. In addition, the errors between the MPM and SPH results \( V_{\text{numerical}} \) and the theoretical values \( V_{\text{theoretical}} \) are quantified by the \( L_2 \) and \( L_\infty \) norms as defined by

\[ L_2 = \sqrt{\sum (V_{\text{theoretical}} - V_{\text{numerical}})^2} \quad (34) \]

\[ L_\infty = \max \left| \frac{V_{\text{theoretical}} - V_{\text{numerical}}}{\max |V_{\text{theoretical}}|} \right| \quad (35) \]

3.1 Poiseuille flow
As shown in Figure 4, a fluid initially at rest between two stationary plates is driven by a uniform body force \( F \) parallel to the \( x \)-axis. The detailed geometrical parameters and material properties are given in Table 1. The MPM background grid is composed of cube-shaped cells with a side length \( L_c = 0.025 \text{ mm} \). Four different MPM models of 800, 3,200, 7,200, and 12,800 points are initialised with 1, 4, 9 and 16 points per grid cell, respectively. This means that one layer of points are set along \( z \)-axis. For comparison, a total of four SPH models are constructed using the initial particle spaces of 0.025, 0.0125, 0.00833, and 0.00625 mm, respectively, and have the same particle numbers as those for the MPM models. Table 2 summarises the detailed MPM and SPH model parameters, including the additional boundary particle number for each SPH model. The periodic boundary conditions are applied at the right and left boundaries in both MPM and SPH, where the particles passing through the right boundary re-enter into the computational domain with the same velocities and state variables through the opposite left boundary. Moreover, the same time step of 5 \( \mu \)s is adopted for all MPM and SPH simulations.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Geometrical parameters and material properties for Poiseuille flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L ) (mm)</td>
<td>( H ) (mm)</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5(a) shows the time-history of the maximum velocity along the \( x \)-axis in the fluid that is obtained by the first 30 terms of the following series solution (Morris et al., 1997)
$V_s(y, t) = \frac{F}{2H} y(y - H) + \sum_{n=0}^{\infty} \frac{4FH^2}{\sqrt{\pi} (2n+1)^3} \times \sin \left( \frac{\pi y}{H} (2n+1) \right) \exp \left[ -\frac{(2n+1)^2 \pi^2 v}{2H^2} t \right] \right] \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \right) \r
Figure 5 (a) Time-history of maximum velocity along the $x$-axis by the series solution of equation (35) and (b) comparisons of the velocity profiles by the MPM and SPH simulations with the theoretical solutions for Poiseuille flow problem (see online version for colours)

Figure 6(a) presents the $L_2$ and $L_\infty$ norms for the MPM and SPH results at $t = 1.0$ s, with the detailed values listed in Table 3. It is evident that the numerical results converge to the theoretical solutions with the increase of particles and that for a given particle number the MPM results show higher accuracy as compared with the SPH solutions. The relationships between computing time per step $T_p$ and total particle number for the MPM and SPH computations are plotted in Figure 6(b), where $T_p$ is calculated by dividing the total CPU time by the total step number of 200,000 (corresponding to $t = 1.0$ s). As expected, it is found that the computational costs for the MPM are much less than those for the SPH. Specifically, the value of $T_p$ for the MPM is about one-tenth of that for the SPH, as detailed listed in Table 3. Table 3 also presents the memory usages of non-swapped physical memory for both methods for Poiseuille flow monitored by the Linux command of TOP. It is shown that the memory storage used for the MPM is much less than that for the SPH. The significant discrepancies in the MPM and SPH computing costs and memory usages are mainly because of no neighbour search involved in the MPM computation. The additional boundary
particles also contribute to the longer SPH computing time. Based on the comparisons in Figures 6(a) and 6(b) and Table 3, it is implied that the MPM is advantageous over the SPH in simulating fluid dynamics problems for its lower computing cost and higher accuracy.

3.2 Couette flow

Figure 7 illustrates an example problem of Couette flow, with the same geometrical configurations and material properties as those in the previously simulated Poiseuille flow problem. The initially stationary fluid begins to flow at time $t > 0$ due to a moving upper plate at a constant velocity $V = 1 \times 10^{-4}$ m/s along the x-axis, and a fixed lower plate. The details of MPM and SPH simulations for Couette flow are identical to those used for Poiseuille flow in Subsection 3.1, including the time step of 5 μs.

**Figure 6** (a) Comparisons of $L_2$ and $L_\infty$ norms for the MPM and SPH simulations for Poiseuille flow problem and (b) relationships between computing time per step $T_p$ and total particle number for the MPM and SPH computations for Poiseuille flow problem (see online version for colours)
Material point method and smoothed particle hydrodynamics simulations of fluid flow problems

Figure 7 Illustration of Couette flow problem (see online version for colours)

![Couette flow illustration](image)

Table 4 Computing time per step $T_p$ and $L_2$ and $L_\infty$ norms and memory usage for the MPM and SPH calculations of the Couette flow

<table>
<thead>
<tr>
<th>Model</th>
<th>Model #</th>
<th>$T_p$ (ms)</th>
<th>$L_2$</th>
<th>$L_\infty$</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM</td>
<td>M1</td>
<td>0.49</td>
<td>1.08%</td>
<td>0.98%</td>
<td>1.8 MB</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>1.84</td>
<td>0.40%</td>
<td>0.38%</td>
<td>2.8 MB</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>4.01</td>
<td>0.26%</td>
<td>0.24%</td>
<td>4.3 MB</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>7.27</td>
<td>0.20%</td>
<td>0.19%</td>
<td>6.5 MB</td>
</tr>
<tr>
<td>SPH</td>
<td>S1</td>
<td>4.72</td>
<td>1.35%</td>
<td>1.22%</td>
<td>29 MB</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>16.6</td>
<td>0.68%</td>
<td>0.62%</td>
<td>30 MB</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>36.9</td>
<td>0.45%</td>
<td>0.41%</td>
<td>35 MB</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>66.2</td>
<td>0.33%</td>
<td>0.31%</td>
<td>37 MB</td>
</tr>
</tbody>
</table>

Figure 8(a) shows the time-history of the velocity at the position of $y = H/2$ along the $x$-axis in the fluid as determined by the first 30 terms of the series solutions by Morris et al. (1997)

$$V_y(y, t) = \frac{V}{H} y + \sum_{n=0}^{\infty} \frac{2V}{n\pi} (-1)^n \sin \left( \frac{n\pi y}{H} \right) \exp \left[ -\frac{n^2 \pi^2 V}{H^2 t} \right]$$

The velocities at time $t = 0.6$, 0.8, 1.0 and 10.0 s are 0.04983, 0.04997, 0.04999, and 0.05 mm/s, respectively. Thus, the flow at $t = 1.0$ s are considered to be in a steady state and the following discussions are presented based on the velocities at 1.0 s. Figure 8(b) compares the distributions of the particle velocities along the $x$-axis by the MPM and SPH with the theoretical solutions, where the velocities are also plotted based on the average velocities of particles at the same $y$ for the models of 3,200 particles. It can be seen that the predictions of MPM and SPH agree excellently with the theoretical solutions. Figure 9(a) plots the $L_2$ and $L_\infty$ norms for the MPM and SPH results at $t = 1.0$ s, with the
corresponding values listed in Table 4. Again, the more the particles are used, the closer to the theoretical solutions the numerical results are. For a given particle number, the MPM results show higher accuracy as compared with the SPH solutions. The relationships between $T_p$ and particle number for the MPM and SPH computations are presented in Figure 9(b). Likewise, the MPM computing cost is also around one-tenth of that for the SPH (see Table 4). Table 4 also presents the memory usages of non-swapped physical memory for both methods for Couette flow monitored by the Linux command of TOP. As expected, the memory usage for the MPM is also much less than that for the SPH. It is distinct that the simulations of Couette flow also demonstrates higher computational efficiency and accuracy of the MPM in fluid dynamics modelling, showing consistence with the comparisons of the MPM and SPH simulations of Poiseuille flow in Subsection 3.1.

**Figure 8** (a) Time-history of velocity at the position of $y = H/2$ along the x-axis by the series solution of equation (36) and (b) comparisons of the velocity profile by the MPM and SPH simulations with the theoretical solutions for Couette flow problem (see online version for colours)
3.3 Water dam break flow

A problem of two-dimensional water dam break flow, as shown in Figure 10(a), is also studied using the MPM and SPH for comparing their efficiency and accuracy in simulating free surface flow. A water column of $0.6 \times 0.6 \, \text{m}^2$ rests at the right corner of an open tank, and the water column begins to collapse at time $t = 0$ due to the gravity ($g = 9.8 \, \text{m/s}^2$). Figure 10(b) illustrates the geometrical configuration of the flow at time $t$. For both MPM and SPH simulations, the rigid wall boundary condition, which means that the particle velocity component normal to the tank walls are set zero while the particles are allowed to freely move in the tangential direction of the tank walls, is applied. In addition, the artificial bulk modulus $K$ and the time step are 0.36 MPa and 50 μs, respectively, and the simulation time period of interest is 0.5 s.

As given in Table 5, the grid cell size of $L_c = 0.02 \, \text{m}$ is used to mesh the background area $L \times H$, and the MPM models are initialised with the point numbers per grid cell of 1, 4, 9 and 16. Four SPH models are built with the particle
spaces of 0.02, 0.01, 0.00667 and 0.005 m, respectively. Table 5 also lists the total particle numbers for all MPM and SPH model and the numbers of boundary particles used in the SPH computation. The MPM and SPH results are here compared with the experimental work by Lobovský et al. (2014).

Table 5  
<table>
<thead>
<tr>
<th>Model</th>
<th>Model #</th>
<th>( L_c ) (m)</th>
<th>Particle number per cell</th>
<th>Particle space (m)</th>
<th>Boundary particle number</th>
<th>Total particle number ((N_p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM M1</td>
<td>0.02</td>
<td>1</td>
<td>0.02</td>
<td>NA</td>
<td>900</td>
<td></td>
</tr>
<tr>
<td>MPM M2</td>
<td>0.02</td>
<td>4</td>
<td>0.01</td>
<td>NA</td>
<td>3,600</td>
<td></td>
</tr>
<tr>
<td>MPM M3</td>
<td>0.02</td>
<td>9</td>
<td>0.00667</td>
<td>NA</td>
<td>8,100</td>
<td></td>
</tr>
<tr>
<td>MPM M4</td>
<td>0.02</td>
<td>16</td>
<td>0.005</td>
<td>NA</td>
<td>14,400</td>
<td></td>
</tr>
<tr>
<td>SPH S1</td>
<td>NA</td>
<td>NA</td>
<td>0.02</td>
<td>420</td>
<td>900</td>
<td></td>
</tr>
<tr>
<td>SPH S2</td>
<td>NA</td>
<td>NA</td>
<td>0.01</td>
<td>840</td>
<td>3,600</td>
<td></td>
</tr>
<tr>
<td>SPH S3</td>
<td>NA</td>
<td>NA</td>
<td>0.00667</td>
<td>1260</td>
<td>8,100</td>
<td></td>
</tr>
<tr>
<td>SPH S4</td>
<td>NA</td>
<td>NA</td>
<td>0.005</td>
<td>1680</td>
<td>14,400</td>
<td></td>
</tr>
</tbody>
</table>

The snapshots for the breaking water dam at different time instants obtained by the simulations of MPM and SPH with 14,400 particles and the experiment are presented in Figure 11. It looks that both MPM and SPH yield snapshots similar to the experimental results. Figure 12(a) compares the simulated \( L(T) \) as a function of \( T \) with those by the experiment, where \( L(T) \) and \( T \) are dimensionless variables used to track the wave front

\[
L(T) = \frac{l(t)}{l_0}
\]

\[
T = t\sqrt{\frac{h_0g}{l_0^2}}
\]

As can be found, the propagations of the wave front simulated by the MPM and the SPH agree very well with each other, and reasonably match the experimental observations. It is also shown that the MPM simulations converges at a faster rate than the SPH calculations with more particles. The discrepancies between the numerical and experimental results for \( L(T) \) might be because the dam gate removal process in the experiment is somewhat different from that in the simulations. The experiment was performed with the gate quickly lifted by an upward force, and the water flow would be still influenced by the gate before it was completely removed. Whereas, the gate was instantaneously removed at time \( T = 0 \) in the simulations, and has no effect on the flow at \( T > 0 \). This could lead to different water column collapse behaviours in the experiments and simulations. In addition, the differences at the later stage \( T > 1.0 \) may be due to the drag force at the bottom wall, which affects the propagation velocity and causes the turbulence near the water front and increases the delay of experiment results (Colagrossi and Landrini, 2003).

![Figure 10](image-url)  
(a) Setup for water dam break and (b) geometrical configuration of water dam break flow at time \( t \) (see online version for colours)
Figure 11  Flow profiles of experiment results (left column), MPM (middle column) and SPH (right column) simulations for the water dam break flow at \( t = 0.32 \) s (top row), \( t = 0.41 \) s (middle row) and \( t = 0.46 \) s (bottom row) (see online version for colours)

Figure 12  (a) Comparisons of the evolution of the wave front among MPM, SPH simulations and experiment result (Lobovský et al., 2014) and (b) computing time per step \( T_p \) as a function of the total particle number in the MPM and SPH calculations for the water dam break problem (see online version for colours)
Figure 12(b) shows the computing time per step $T_p$ for both SPH and MPM plotted as a function of the particle number. Higher computational efficiency is again observed for the MPM, with the ratios of the $T_p$ value for the SPH to the one for the MPM ranging from ~2.6 to ~4.6, which are significantly reduced as compared with those in the Poiseuille and Couette flows. Such reductions of the differences between the computational costs for the SPH and MPM could be because much more empty MPM background cells exist in the dam-breaking flow problem and thus additional computational expense are incurred. With the increasing particles, the proportion of computational cost spent for empty MPM background cells is decreasing, and $T_p$ for the MPM rises at a much slower rate than that for the SPH, as demonstrated in Figure 12(b). It is noted that we define a MPM background mesh in the whole domain of $L \times H$. Thus, the computational efficiency of the MPM could be further enhanced by using an adaptive background mesh scheme (Lian et al., 2012, 2015).

4 Conclusions

Three representative example problems, namely, Poiseuille and Couette flows of viscous fluids and a more complex free surface flow of water dam break, have been numerically studied by the MPM and SPH, to demonstrate and compare the capabilities of these two meshless particle methods in simulating fluid flows. Both methods present the predictions in good agreements with the analytical solutions and the experimental results. Since the discrete governing equations are solved in the Eulerian background grid mesh, the MPM does not require the time-consuming and memory consumption neighbour particle search and additional boundary particles, which, however, are compulsory in the SPH. This leads to significantly higher computational efficiency of the MPM, as illustrated in the example problems. Moreover, the comparisons also show faster convergence and higher numerical accuracy for the MPM. Therefore, the MPM appears to be a promising and efficient numerical tool for modelling fluid flows. It is shown in this work that the differences in the computing time between the MPM and the SPH is much reduced for the water dam break problem in comparison with those for Poiseuille and Couette flows. This is due to the fact that the empty MPM background cells, in which no material points are housed, cause additional computational costs. The efficiency of the MPM simulations for fluid flow problems could be further enhanced by using an adaptive mesh scheme. It is suggested that future efforts be made to improve the accuracy of MPM in predicting the flow pressure, especially for the complex free surface flows.

Acknowledgements

This work was supported in part by the NSFC (Grant Nos. 11102185 and 11532011), HQL and HTL are also funded by the NSFC (No. 51476150) and Postgraduate innovation Foundation of Shanxi Province (No. 2015BY45).

References


Material point method and smoothed particle hydrodynamics simulations of fluid flow problems


