

Multi-phase fluid flow simulation by using peridynamic differential operator

Yan Gao and Selda Oterkus*

Department of Naval Architecture, Ocean and Marine Engineering

University of Strathclyde, 100 Montrose Street, Glasgow, G4 0LZ, United Kingdom

Abstract

The problems of multi-phase fluid flows are often encountered in engineering. In this study, a non-local numerical model of multi-phase fluid flows in the Lagrangian description is developed. Based on the peridynamic theory, a peridynamic differential operator is proposed which can convert any arbitrary order of differentials into their integral form without calculating the peridynamic parameters. Therefore, the Navier-Stokes equations including the surface tension forces are reformulated into their integral form. Subsequently, an updated Lagrangian algorithm for solving the multi-phase fluid flow problems is proposed. Besides, the particle shifting technology and moving least square algorithm are also adopted to avoid the possible tension instability. Finally, several benchmark multi-phase fluid flow problems such as two-phase hydrostatic problem, two-phase Poiseuille flow, and 2D square droplet deformation are solved to validate the proposed non-local model. It can be concluded from the current study that the peridynamic differential operator can be applied as an alternative method for multi-phase fluid flow simulation.

Key Words: peridynamic differential operator, non-local model, multi-phase flows, N-S equations, surface tension force

1 Introduction

Multi-phase fluid flow is a typical phenomenon in the engineering fields, such as oil and water flow in a pipe, gas-liquid flow in channels, combustions (Turns, 1996) etc.. Therefore, the problem of the multi-phase fluid flow has been intensively and extensively investigated for years.

*Corresponding Author: Selda Oterkus, Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde. Email: selda.oterkus@strath.ac.uk.

For Euler grid-based methods, e.g. the volume of fluid (VOF) (Hirt and Nichols, 1981) and level set (LS) method (van der Pijl et al., 2005), the unphysical re-initialization process is needed for large topological deformation (Yang et al., 2019). Compared with the Euler methods, the Lagrangian particle methods, e.g. smoothed particle hydrodynamics (SPH) (Cummins and Rudman, 1999; Monaghan and Rafiee, 2013; Tartakovsky and Meakin, 2005; Wang et al., 2016), moving particle semi-implicit (MPS) method (Guo et al., 2018; Koshizuka and Oka, 2017; Wang and Zhang, 2019), can simultaneously show and move the multi-phase fluid interface because of their intrinsic non-local property. Therefore, these particle methods have also been widely applied in the realm of the multi-phase fluid flow simulation. Taking the SPH method as an example, Monaghan and Rafiee (Monaghan and Rafiee, 2013) developed a simple SPH algorithm for multi-fluid flow with introducing a repulsive force on the interface to maintain the interface sharpness. The Incompressible SPH (ISPH) scheme is applied in (Rezavand et al., 2018) with a velocity divergence-free projection algorithm to solve the benchmark multi-fluid problems. The comparative study has been conducted for multi-phase fluid flow to estimate the available SPH techniques (Fürstenau et al., 2016; Suresh et al., 2019b). The state-of-art SPH application for complex fluid flows is reviewed in (TingYe et al., 2019; Wang et al., 2016).

Peridynamics (PD), which is another particle method, is proposed by Silling in Lagrangian approach (Silling, 2000, 2010) originally for the discontinuity issues in solid mechanics. The classical equation of motion is reformulated from the spatial differential form into a spatial integral form. Then for the discretized material body, a particle can interact with other particles within a predefined distance. With about twenty years' development, the PD theory has been intensively employed in the solid mechanical field (Madenci and Oterkus, 2014; Oterkus and Madenci, 2011, 2012; Oterkus et al., 2012; Oterkus and Madenci, 2015, 2017), diffusion field (Bobaru and Duangpanya, 2010; Diyaroglu et al., 2017; Diyaroglu et al., 2018; Han et al., 2015; Nguyen and Oterkus, 2019; Oterkus et al., 2014b), and other multi-physics applications (Gao and Oterkus, 2018, 2019a; Liu et al., 2019; Liu et al., 2018; Nguyen and Oterkus, 2019; Oterkus, 2015; Oterkus et al., 2013; Oterkus et al., 2014a; Oterkus et al., 2017; Oterkus et al., 2014c; Wang et al., 2018b). Based on the PD theory, a peridynamic differential operator (PDDO) is proposed by Madenci et al. (Madenci et al., 2016). The PDDO is a non-local operator which can express the spatial partial derivatives in an integral form without any limitation on the differential orders. Also, the classical material physical parameters are directly utilized in the integral expressions (Madenci et al., 2019), leading to a simple implementation. The developed

PDDO has been coupled with different methods. PDDO is coupled with the finite difference method to solve the acoustic wave propagation problems in infinite and semi-infinite domains (Shojaei et al., 2019). It is also coupled with the refined zigzag theory (Tessler et al., 2009) for stress analysis of laminated composite beam under different loading conditions (Dorduncu, 2019). The PDDO is also applied to predict the crack propagation during the lithiation process (Wang et al., 2018a, 2018b, 2018c). Later on, PDDO model is developed to solve fluid flow problems (Bazazzadeh et al., 2018; Gao and Oterkus, 2019b, 2019c). Therefore, in this study PDDO is adopted. The PD application on fluid mechanics has drawn the interest of the researchers recently. Silling et al. (Silling et al., 2017) developed the PD in an Euler approach for modelling shockwaves and impact phenomenon. Oterkus et al. (Oterkus et al., 2017) simulated the fracture in fluid-filled structures by using a fully coupled poroelastic PD formulation. Updated Lagrangian particle hydrodynamics (ULPH) is proposed by Tu and Li (Tu and Li, 2017) by using the non-ordinary state-based PD concept and updated Lagrangian scheme. Subsequently, the ULPH has been extended to simulate the multi-phase fluid flow (Yan et al., 2019). Wang and Zhang developed a PD-Moving Particle Semi-implicit (MPS) model for multi-phase fluid flow simulations (Wang and Zhang, 2019). The PDDO has been applied to simulate the laminar fluid flow at low Reynold number (Gao and Oterkus, 2019c) and heat-conducting fluid flow (Gao and Oterkus, 2019b). In the present work, the usage of PDDO in single-phase fluid flows is extended for multi-phase fluid flows with low density ratios at low Reynold numbers.

This paper is organized as follows. Section 2 briefly reviews the idea of PDDO. Section 3 describes the non-local PDDO model for multi-phase fluid flow and the time integration scheme. Section 4 provides information regarding numerical implementations. Section 5 presents the numerical simulations for benchmark multi-phase fluid flows. Finally, Section 6 presents the conclusion.

2 Basics of Peridynamic Differential Operator

Peridynamic differential operator (PDDO), which is based on the basic idea of Peridynamic (PD) theory (Silling, 2000) of using integral equations instead of differential equations, is recently proposed (Madenci et al., 2016). The PDDO concepts based on the PD theory (Silling, 2000) are briefly reviewed in this section.

The PDDO can be used to calculate the values of any order partial differentials by using integral formulations (Madenci et al., 2016). The derivatives of a scalar function $f(\mathbf{x})$ up to second-order derivatives in two-dimensional space can be calculated as (Madenci et al., 2016)

$$\begin{bmatrix} \partial f(\mathbf{x}) / \partial x_1 \\ \partial f(\mathbf{x}) / \partial x_2 \\ \partial^2 f(\mathbf{x}) / \partial x_1^2 \\ \partial^2 f(\mathbf{x}) / \partial x_2^2 \\ \partial^2 f(\mathbf{x}) / \partial x_1 \partial x_2 \end{bmatrix} = \int_{H_x} (f(\mathbf{x}') - f(\mathbf{x})) \begin{bmatrix} g^{10}(\xi) \\ g^{01}(\xi) \\ g^{20}(\xi) \\ g^{02}(\xi) \\ g^{11}(\xi) \end{bmatrix} dV' \quad (1)$$

where $g^{10}(\xi)$, $g^{01}(\xi)$, $g^{20}(\xi)$, $g^{02}(\xi)$, $g^{11}(\xi)$ represents the peridynamic operators and ξ represents the relative position between material point \mathbf{x} and its family member \mathbf{x}' as

$$\xi = \mathbf{x}' - \mathbf{x} = \xi_1 \mathbf{e}_1 + \xi_2 \mathbf{e}_2 \quad (2)$$

with

$$\xi_1 = x'_1 - x_1; \xi_2 = x'_2 - x_2 \quad (3)$$

The terms ξ_1 and ξ_2 represent the projections of the vector ξ on the coordinate axis with unit vectors being \mathbf{e}_1 , \mathbf{e}_2 . V' represents the volume of point \mathbf{x}' . H_x is the horizon of point \mathbf{x} (Silling, 2000).

Eq.(1) can be also represented in compact form as

$$\frac{\partial^{p_1+p_2} f(\mathbf{x})}{\partial x_1^{p_1} \partial x_2^{p_2}} = \int_{H_x} (f(\mathbf{x} + \xi) - f(\mathbf{x})) g^{p_1 p_2}(\xi) dV' \quad (4)$$

with

$$0 < p_1 + p_2 \leq 2 \quad (5)$$

where the PDDO is denoted by $g^{p_1 p_2}(\xi)$ with the integer variables p_1 , and p_2 representing the differential orders with respect to x_1 and x_2 , respectively.

The peridynamic operators in Eq.(1) can be represented as (derivation is provided in Appendix A) (Madenci et al., 2016)

$$\begin{bmatrix} g^{10}(\xi) \\ g^{01}(\xi) \\ g^{20}(\xi) \\ g^{02}(\xi) \\ g^{11}(\xi) \end{bmatrix} = \left(\mathbf{A}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)^T \begin{bmatrix} w\xi_1 \\ w\xi_2 \\ w\xi_1^2 \\ w\xi_2^2 \\ w\xi_1\xi_2 \end{bmatrix} \quad (6a)$$

$$\mathbf{A} = \left(\int_{H_x} w \begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1^3 & \xi_1\xi_2^2 & \xi_1^2\xi_2 \\ \xi_1\xi_2 & \xi_2^2 & \xi_1^2\xi_2 & \xi_2^3 & \xi_1\xi_2^2 \\ \xi_1^3 & \xi_1^2\xi_2 & \xi_1^4 & \xi_1^2\xi_2^2 & \xi_1^3\xi_2 \\ \xi_1\xi_2^2 & \xi_2^3 & \xi_1^2\xi_2^2 & \xi_2^4 & \xi_1\xi_2^3 \\ \xi_1^2\xi_2 & \xi_1\xi_2^2 & \xi_1^3\xi_2 & \xi_1\xi_2^3 & \xi_1^2\xi_2^2 \end{bmatrix} dV' \right) \quad (6b)$$

where w represents the weight function as (Madenci et al., 2016)

$$w = e^{-(2|\xi|/\delta)^2} \quad (7)$$

It should be noted that since the second-order Taylor series expansion is adopted, the horizon size should be chosen as $\delta = m\Delta x$ with $2 \leq m \leq 4$ and Δx representing the particle spacing (Madenci et al., 2019).

Furthermore, the PDDO can be categorized into two matrices, i.e. a vector for the first-order derivatives

$$\mathbf{g}_1(\xi) = \begin{bmatrix} g^{10}(\xi) & g^{01}(\xi) \end{bmatrix} \quad (8)$$

and a symmetric matrix for the second-order derivatives

$$\mathbf{g}_2(\xi) = \begin{bmatrix} g^{20}(\xi) & g^{11}(\xi) \\ g^{11}(\xi) & g^{02}(\xi) \end{bmatrix} \quad (9)$$

Consequently, the first-order and second-order partial derivatives for a vector function $\mathbf{f}(\mathbf{x})$ can be calculated through the PDDO as (Gao and Oterkus, 2019b, c)

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = \int_{H_x} \mathbf{g}_1(\xi) \cdot (\mathbf{f}(\mathbf{x}') - \mathbf{f}(\mathbf{x})) dV'; \quad (10a)$$

$$\nabla \mathbf{f}(\mathbf{x}) = \int_{H_x} (\mathbf{f}(\mathbf{x}') - \mathbf{f}(\mathbf{x})) \otimes \mathbf{g}_1(\xi) dV'; \quad (10b)$$

$$\Delta \mathbf{f}(\mathbf{x}) = \int_{H_{\mathbf{x}}} \text{tr}(\mathbf{g}_2(\xi))(\mathbf{f}(\mathbf{x}') - \mathbf{f}(\mathbf{x}))dV'; \quad (10c)$$

where Δ represents the Laplacian operator. In this study, the PDDO is utilized to reformulate the governing equations for multi-phase fluids modelling in a non-local fashion. The discretised form of Eq. (10) is provided for point of interested \mathbf{x}_i and its family member \mathbf{x}_j as

$$\nabla \cdot \mathbf{f}(\mathbf{x}_i) = \sum_{i=1}^{N_i} \mathbf{g}_1(\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{x}_i))V_j \quad (11a)$$

$$\nabla \mathbf{f}(\mathbf{x}_i) = \sum_{i=1}^{N_i} (\mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{x}_i)) \otimes \mathbf{g}_1(\mathbf{x}_j - \mathbf{x}_i)V_j \quad (11b)$$

$$\Delta \mathbf{f}(\mathbf{x}_i) = \sum_{i=1}^{N_i} \text{Tr}(\mathbf{g}_2(\mathbf{x}_j - \mathbf{x}_i))(\mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{x}_i))V_j \quad (11c)$$

3 Governing Equations for Multiphase Flow

In this study, the fluids are assumed to be weakly compressible, viscous, immiscible, Newtonian fluids under an isothermal laminar flow condition. The flow governing equations are reformulated to a Lagrangian form by using the peridynamic differential operator.

3.1 Classical governing equations

The governing equations for each fluid phase in the Lagrangian description are (Zhang et al., 2015)

Continuity equation:

$$\frac{\partial \rho}{\partial t} = -\rho \nabla \cdot \mathbf{v} \quad (12)$$

Momentum equation:

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla P + \mathbf{F}^v + \mathbf{F}^B + \mathbf{F}^S \quad (13)$$

where the variables are denoted as density ρ , time t , fluid velocity vector \mathbf{v} , pressure P , viscous stress \mathbf{F}^v , body force \mathbf{F}^B , and surface tension force \mathbf{F}^S .

The viscous stress \mathbf{F}^v is defined as (Zhang et al., 2015)

$$\mathbf{F}^v = \mu(\Delta \mathbf{v}) \quad (14)$$

where μ represents the dynamic viscosity.

According to the Continuum Surface Force model (Brackbill et al., 1992), the surface tension force is defined as

$$\mathbf{F}^s = \beta \kappa \hat{\mathbf{n}} \lambda + \nabla \beta \quad (15)$$

where β is the surface tension coefficient, $\hat{\mathbf{n}}$ is the unit normal towards the interface as shown in Fig. 1, κ is the curvature of the interface, and λ is the weight function representing the surface tension force magnitude distribution. The surface tension coefficient is set as a constant value. Therefore, the second term on the right-hand side of Eq. (15) which is the gradient of the surface tension coefficient becomes zero, i.e.

$$\nabla \beta = 0 \quad (16)$$

According to the Continuum Surface Force model (Brackbill et al., 1992), a colour function, s , is given to fluid A and B to identify the different phases (shown in different colours in Fig. 1). For example,

$$s = \begin{cases} 1, & \text{for fluid A} \\ 2, & \text{for fluid B} \end{cases} \quad (17)$$

Then, the gradient of the colour function representing the direction of the interface can be calculated as

$$\mathbf{c}_g = \nabla s \quad (18)$$

Consequently, the unit normal direction vector, $\hat{\mathbf{n}}$ in Eq.(15), can be calculated based on the normal direction vector provided in Eq.(18) as

$$\hat{\mathbf{n}} = \frac{\mathbf{c}_g}{|\mathbf{c}_g|} = \frac{\nabla s}{|\nabla s|} \quad (19)$$

The curvature of the interface in Eq. (15) can be calculated as the negative value of the divergence of the unit normal vector (Brackbill et al., 1992), i.e.

$$\kappa = -\nabla \cdot \hat{\mathbf{n}} \quad (20)$$

The weighted function λ in Eq. (15) represents the distribution of the magnitude of the surface tension force (Zhang et al., 2015) (the magnitude decays as the distance to interface increases). Following the conventional choice, the weighted function is chosen as the magnitude of the gradient of the colour function as (Zhang et al., 2015)

$$\lambda = |\mathbf{c}_g| = |\nabla s| \quad (21)$$

As a result, by substituting Eq.(16), Eq.(19), Eq.(20), and Eq.(21) into Eq.(15), the surface tension force can be expressed as

$$\mathbf{F}^S = \beta \kappa \hat{\mathbf{n}} \lambda + \nabla \beta = \beta \kappa \nabla s = \beta \left(-\nabla \cdot \left(\frac{\nabla s}{|\nabla s|} \right) \right) \nabla s \quad (22)$$

The above governing equations are closed with the equation of state. In the current work, two-phase fluid flows are considered, denoted as phase A for heavier fluid and phase B for lighter fluid. Therefore, for each fluid phase (denoted as a subscript), the equation of state can be expressed as (Zhang et al., 2015)

$$P_A = P_{ref,A} \left(\left(\frac{\rho_A}{\rho_{0,A}} \right)^{\gamma_A} - 1 \right) + P_{0,A} \quad (23a)$$

$$P_B = P_{ref,B} \left(\left(\frac{\rho_B}{\rho_{0,B}} \right)^{\gamma_B} - 1 \right) + P_{0,B} \quad (23b)$$

where $\rho_{0,A}$ and $\rho_{0,B}$ are the initial density of fluid A and B, γ_A and γ_B are the material constants for fluid A and B ranging from 1 to 7. The terms $P_{ref,A}$ and $P_{ref,B}$ are the reference pressures as (Zhang et al., 2015)

$$P_{ref,A} = \frac{\rho_{0,A} c_A^2}{\gamma_A} \quad (24a)$$

$$P_{ref,B} = \frac{\rho_{0,B} c_B^2}{\gamma_B} \quad (24b)$$

where c_A and c_B are the artificial speeds of sound for fluid A and B, respectively. For both fluid phases, the reference pressures are set to be equal, i.e. $P_{ref,A} = P_{ref,B} = P_{ref}$. Besides, the artificial speed of sound for fluid A, c_A , is chosen as the reference speed of sound. Therefore, the artificial speed of sound for fluid B, c_B , is calculated as $c_B = c_A \sqrt{(\gamma_B \rho_{0,A} / \gamma_A \rho_{0,B})}$ (Zhang et al., 2015).

The term $P_{0,A}$ or $P_{0,B}$ in Eq.(22) is referred to as the background pressure to prevent negative pressure and tension instabilities (Krimi et al., 2018; Zhang et al., 2015). The adoption of the background pressure can keep the particle distribution uniformly to some extent. Therefore, it should not be too large to create extra particle resettlement and thus induce numerical simulation errors (Colagrossi et al., 2012). For both fluids, the background pressure is chosen to be identical as $P_{0,A} = P_{0,B} = P_0 = 0.05 P_{ref}$ for problems with no free surface flow and $P_{0,A} = P_{0,B} = 0$ for free surface flow problems (Krimi et al., 2018).

3.2 PDDO governing equations

The non-local mathematical model is developed by reformulating the classical mathematical model by using PDDO.

The non-local form of the continuity equation, the pressure gradient term and the viscous force term in the momentum equation remain the same as in the single-phase flow as provided by Gao and Oterkus (Gao and Oterkus, 2019c), and they can be presented as follows:

The non-local form of the continuity equation, i.e. Eq.(12):

$$\frac{\partial \rho(\mathbf{x})}{\partial t} = -\rho(\mathbf{x}) \int_{H_x} \mathbf{g}_1(\xi) \cdot (\mathbf{v}(\mathbf{x}') - \mathbf{v}(\mathbf{x})) dV' \quad (25)$$

The non-local form of the pressure gradient in the momentum equation presented in Eq.(13):

$$\nabla P(\mathbf{x}) = \int_{H_x} (P(\mathbf{x}') - P(\mathbf{x})) \mathbf{g}_1(\xi) dV' \quad (26)$$

The non-local form of viscous force in the momentum equation presented in Eq.(14):

$$\mathbf{F}^v = \int_{H_x} \mu(\mathbf{x}) (\mathbf{v}(\mathbf{x}') - \mathbf{v}(\mathbf{x})) \text{Tr}(\mathbf{g}_2(\xi)) dV' \quad (27)$$

The derivation of the non-local form of surface tension force is presented as follows:

In order to develop the PDDO form of the surface tension force model, the gradient of the colour function, ∇s , should be firstly expressed by using the PDDO formulation as

$$\nabla s(\mathbf{x}) = \int_{H_x} (s(\mathbf{x}') - s(\mathbf{x})) \mathbf{g}_1(\xi) dV' \quad (28)$$

If the density difference is considerable, as an alternative choice for the colour function difference $(s(\mathbf{x}') - s(\mathbf{x}))$ in Eq.(28), the interparticle colour index number (Adami et al., 2010) provided in Eq. (29) can also be utilized. Hence, the continuity of the acceleration across the interface is guaranteed by considering the density difference (Zhang et al., 2015). Therefore, in this study, Eq. (29) is used for the surface tension force calculation.

$$(s(\mathbf{x}') - s(\mathbf{x})) = \mathbb{C}(\mathbf{x}', \mathbf{x}) = \begin{cases} \frac{2\rho_{x,0}}{\rho_{x,0} + \rho_{x',0}}, & \text{if } \mathbf{x} \text{ and } \mathbf{x}' \text{ belong to different phases} \\ 0, & \text{if } \mathbf{x} \text{ and } \mathbf{x}' \text{ belong to the same phase} \end{cases} \quad (29)$$

and Eq.(28) becomes (Adami et al., 2010)

$$\nabla s(\mathbf{x}) = \int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \quad (30)$$

where $\rho_{x,0}$ and $\rho_{x',0}$ represent the initial densities of particle \mathbf{x} and \mathbf{x}' , respectively. As a result, the colour function gradient in Eq. (18) becomes

$$\mathbf{c}_g(\mathbf{x}) = \nabla s(\mathbf{x}) = \int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \quad (31)$$

Consequently, the weighted function in Eq. (21) which is equal to the magnitude of the normal direction vector becomes

$$\lambda = |\mathbf{c}_g(\mathbf{x})| = |\nabla s| = \left| \int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \right| \quad (32)$$

and the unit normal vector $\hat{\mathbf{n}}$ Eq.(19) hence can be calculated as

$$\hat{\mathbf{n}}(\mathbf{x}) = \frac{\mathbf{c}_g(\mathbf{x})}{|\mathbf{c}_g(\mathbf{x})|} = \frac{\nabla s}{|\nabla s|} = \frac{\int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV'}{\left| \int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \right|} \quad (33)$$

The curvature $\kappa(\mathbf{x})$ in Eq.(20) thus can be calculated by the PDDO formulation as

$$\kappa^*(\mathbf{x}) = -\nabla \cdot \hat{\mathbf{n}}(\mathbf{x}) = -\int_{H_x} (\hat{\mathbf{n}}(\mathbf{x}') - \hat{\mathbf{n}}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV' \quad (34)$$

According to the explanation given in (Morris, 2000), the normal vector \mathbf{n} may have erroneous direction and small value away from the interface, therefore a cut-off value ($\varepsilon = 1.0 \times 10^{-2} / \Delta x$ as suggested in (Morris, 2000)) is set for the normal vector $\mathbf{c}_g(\mathbf{x})$, smaller than which the unit normal vector $\hat{\mathbf{n}}(\mathbf{x})$ is set to be zero. A function $\mathbb{N}(\mathbf{x})$ is defined to indicate if the unit normal vector is zero as

$$\mathbb{N}(\mathbf{x}) = \begin{cases} 1, & \text{if } |\mathbf{c}_g(\mathbf{x})| > \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (35)$$

And the calculation of the unit normal vector $\hat{\mathbf{n}}(\mathbf{x})$ is set as

$$\hat{\mathbf{n}}(\mathbf{x}) = \begin{cases} \mathbf{c}_g(\mathbf{x}) / |\mathbf{c}_g(\mathbf{x})|, & \text{if } \mathbb{N}(\mathbf{x}) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (36)$$

Therefore, the PDDO form of the unit normal direction $\hat{\mathbf{n}}(\mathbf{x})$ for particle \mathbf{x} in Eq. (33) becomes as

$$\hat{\mathbf{n}}(\mathbf{x}) = \frac{\mathbf{c}_g(\mathbf{x})}{|\mathbf{c}_g(\mathbf{x})|} = \frac{\nabla S}{|\nabla S|} = \begin{cases} \frac{\int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV'}{\left| \int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \right|}, & \text{if } \mathbb{N}(\mathbf{x}) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (37)$$

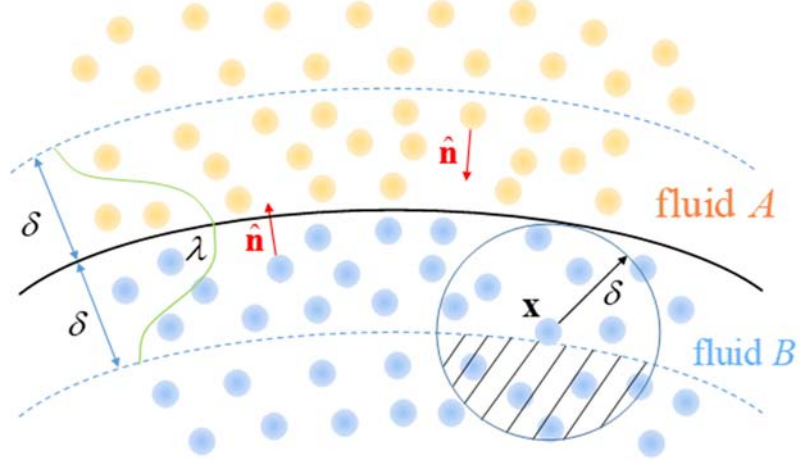


Fig. 1 Surface tension force in the multi-fluid interface region

The approach for curvature calculation provided in (Morris, 2000) is adopted to increase the approximation accuracy. An intermediate estimate of the curvature (the divergence of the unit normal vector $\nabla \cdot \hat{\mathbf{n}}$) is defined using a sum over reliable neighbouring unit normal vectors as (Morris, 2000)

$$\kappa^{**}(\mathbf{x}) = -(\nabla \cdot \hat{\mathbf{n}}(\mathbf{x}))^* = -\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) (\hat{\mathbf{n}}(\mathbf{x}') - \hat{\mathbf{n}}(\mathbf{x})) \cdot \mathbf{g}_i(\xi) dV' \quad (38)$$

The width of the multi-fluid interface region is defined as 2δ , as shown in Fig. 1. It can be referred from Fig. 1 that if the central point \mathbf{x} is located at the edge of the interface transition region, the unit normal vectors of its family members located in the shaded region in Fig. 1 are forced to be zero. In other words, the neighbourhood of the interface region boundary particle is truncated. Being similar to surface correction for the boundary particles in peridynamic discretization and the approach adopted in (Morris, 2000), a correction factor for $\kappa^{**}(\mathbf{x})$ is proposed as

$$\zeta^*(\mathbf{x}) = \frac{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) w_0(\xi) dV'}{\int_{H_x} w_0(\xi) dV'} \quad (39)$$

where w_0 represents the improved Gaussian weighted function as (Colagrossi and Landrini, 2003)

$$w_0(|\xi|) = \frac{1}{H} \left(e^{-\frac{(|\xi|/h)^2}{10}} - e^{-9} \right) / \left(h^2 \pi (1 - 10e^{-9}) \right), |\xi| < \delta, \text{ for 2D} \quad (40)$$

In Eq.(40), h represents the smoothing length (Colagrossi and Landrini, 2003) with its value being set as $h = 1.2\Delta x$. The thickness is denoted by H with the volume of one particle being calculated as $V = (\Delta x)^2 H$ in a uniform mesh size. In the present study, a unit thickness is adopted for 2D simulations. The term $\zeta^*(\mathbf{x})$ can reflect the number density of particle \mathbf{x} (Morris, 2000). As a result, the curvature $\kappa(\mathbf{x})$ in Eq.(34) is modified as (Morris, 2000)

$$\kappa^{***}(\mathbf{x}) = \frac{\kappa^{**}(\mathbf{x})}{\zeta^*(\mathbf{x})} = - \frac{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) (\hat{\mathbf{n}}(\mathbf{x}') - \hat{\mathbf{n}}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV'}{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) w_0(\xi) dV' / \int_{H_x} w_0(\xi) dV'} \quad (41)$$

As shown in Fig. 1, it should be noted that the unit normal $\hat{\mathbf{n}}$ always points from the fluid phase itself to the other fluid phase. Consequently, for the calculation of the curvature, $\kappa(\mathbf{x})$ for central point \mathbf{x} , the unit normal direction vector $\hat{\mathbf{n}}(\mathbf{x}')$ for the family member \mathbf{x}' is inverted if point \mathbf{x}' and point \mathbf{x} belong to different fluid phases. Hence, Eq.(41) is modified as

$$\kappa(\mathbf{x}) = - \frac{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) (\varphi \hat{\mathbf{n}}(\mathbf{x}') - \hat{\mathbf{n}}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV'}{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) w_0(\xi) dV' / \int_{H_x} w_0(\xi) dV'} \quad (42)$$

with an additional parameter being added as (Zhang et al., 2015)

$$\varphi = \begin{cases} -1, & \text{if } \mathbf{x} \text{ and } \mathbf{x}' \text{ belong to different phases} \\ 1, & \text{if } \mathbf{x} \text{ and } \mathbf{x}' \text{ belong to the same phases} \end{cases} \quad (43)$$

Finally, the surface tension force is expressed in PDDO form as

$$\begin{aligned} \mathbf{F}^S &= \beta \kappa \nabla s \\ &= -\beta \frac{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) (\varphi \hat{\mathbf{n}}(\mathbf{x}') - \hat{\mathbf{n}}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV'}{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) w_0(\xi) dV' / \int_{H_x} w_0(\xi) dV'} \left(\int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \right) \end{aligned} \quad (44)$$

In conclusion, the non-local form of the momentum equation Eq.(13) becomes

$$\begin{aligned} \rho(\mathbf{x}) \frac{\partial v(\mathbf{x})}{\partial t} = & - \int_{H_x} ((P(\mathbf{x}') - P(\mathbf{x})) \mathbf{g}_1(\xi)) dV' + \int_{H_x} (\mu(\mathbf{x})(\mathbf{v}(\mathbf{x}') - \mathbf{v}(\mathbf{x})) \text{Tr}(\mathbf{g}_2(\xi))) dV' \\ & - \beta \frac{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) (\varphi \hat{\mathbf{n}}(\mathbf{x}') - \hat{\mathbf{n}}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV'}{\int_{H_x} \min(\mathbb{N}(\mathbf{x}), \mathbb{N}(\mathbf{x}')) w_0(\xi) dV' / \int_{H_x} w_0(\xi) dV'} \left(\int_{H_x} \mathbb{C}(\mathbf{x}', \mathbf{x}) \mathbf{g}_1(\xi) dV' \right) \\ & + \mathbf{F}^B \end{aligned} \quad (45)$$

4 Numerical Implementation for Multiphase Fluid Simulation

4.1 Discretised form of PDDO equations

In Fig. 2, \mathbf{x}_i^n represents the current coordinate of a material point at $t = t_n$ in a material body R . Since in PD theory the long-range force is considered, the neighbourhood within which the central point \mathbf{x}_i^n can interact with other points is called horizon and denoted as $H_{\mathbf{x}_i^n}$. The radius of $H_{\mathbf{x}_i^n}$, called horizon size and denoted by δ , is considered constant. The material points inside the horizon are called family members of point \mathbf{x}_i^n . The behaviour of point \mathbf{x}_i^n is determined by the interactions with its family members (Madenci and Oterkus, 2014).

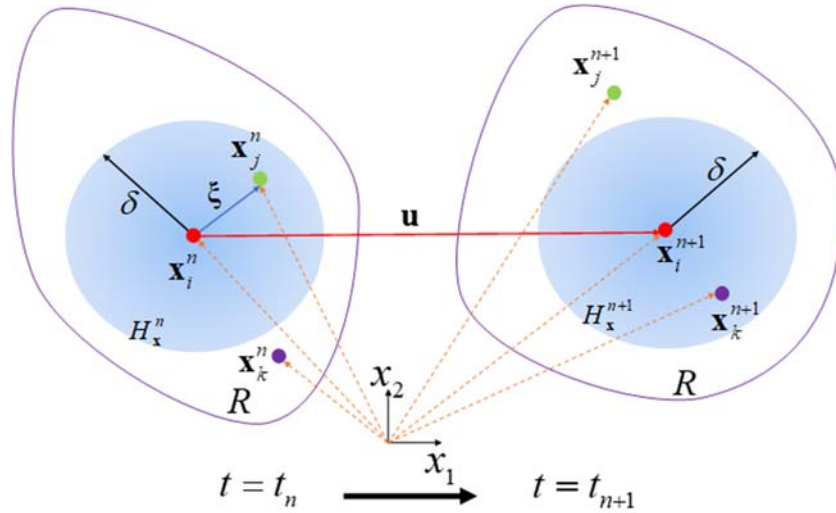


Fig. 2 Interaction of a material point, \mathbf{x}_i^n and its family members at $t = t_n$ and $t = t_{n+1}$

Since the updated Lagrangian description is adopted, the family members of each material point are updated at each time step. For example, the location of a material point is \mathbf{x}_i^n at $t = t_n$ and \mathbf{x}_i^{n+1} at $t = t_{n+1}$. The material point \mathbf{x}_j^n is one of the family members of \mathbf{x}_i^n , on the contrary, material point \mathbf{x}_k^n is located outside the neighbourhood of point \mathbf{x}_i^n , at $t = t_n$. However, at time $t = t_{n+1}$, material point \mathbf{x}_j^{n+1} moves outside the horizon and material point \mathbf{x}_k^{n+1} becomes one of

the family members of point \mathbf{x}_i^{n+1} . In Fig. 2, $H_{\mathbf{x}_i}^n$ and $H_{\mathbf{x}_i}^{n+1}$ represent the horizon at $t = t_n$ and $t = t_{n+1}$, respectively.

In the numerical simulation, the updated Lagrangian description is adopted, as explained in (Gao and Oterkus, 2019c). Therefore, the reference configuration is referred to as the current configuration. The PDDO is reconstructed every time step based on the updated family member array. Correspondingly, the integral governing equations can also be expressed by using the discretized particles. The central point is denoted as i and its family member is denoted as j . When fluid particle i is outside the interface region, the continuity equation Eq.(25) is discretized as

$$\rho_i^{n+1} = \rho_i^n - \rho_i^n \Delta t \sum_{j=1}^{N_i} [(\mathbf{v}_j^n - \mathbf{v}_i^n) \cdot \mathbf{g}_1(\boldsymbol{\xi}_{ij}^n)] V_j \quad (46a)$$

with

$$\boldsymbol{\xi}_{ij}^n = \mathbf{x}_j^n - \mathbf{x}_i^n \quad (46b)$$

where N_i represents the number of family members of point i and $\boldsymbol{\xi}_{ij}^n$ represents the relative position between particle i and j in the current configuration $t = t_n$. The volume of particle, V_j is calculated based on the mass and the density of particle, ρ_j^n .

The PDDO form of momentum equation Eq.(45) for fluid particle i which is outside the interface region becomes

$$\mathbf{a}_i^{n+1} = \frac{1}{\rho_i^{n+1}} \left(\begin{array}{l} -\sum_{j=1}^{N_i} (P_j^n - P_i^n) \mathbf{g}_1(\boldsymbol{\xi}_{ij}^n) V_j + \sum_{j=1}^{N_i} \mu_i (\mathbf{v}_j^n - \mathbf{v}_i^n) \text{Tr}(\mathbf{g}_2(\boldsymbol{\xi}_{ij}^n)) V_j \\ -\beta \frac{\left(\sum_{j=1}^{N_i} \min(N_i, N_j) (\varphi \hat{\mathbf{n}}_j^n - \hat{\mathbf{n}}_i^n) \cdot \mathbf{g}_1(\boldsymbol{\xi}_{ij}^n) V_j \right)}{\sum_{j=1}^{N_i} \min(N_i, N_j) w_0(\boldsymbol{\xi}_{ij}^n) V_j / \sum_{j=1}^{N_i} w_0(\boldsymbol{\xi}_{ij}^n) V_j} \left(\sum_{j=1}^{N_i} \mathbf{C}_{ij} \mathbf{g}_1(\boldsymbol{\xi}_{ij}^n) V_j \right) \end{array} \right) + \frac{1}{\rho_i^{n+1}} \mathbf{F}^B \quad (47)$$

where \mathbf{a}_i^{n+1} represents the acceleration of particle i in the updated configuration at $t = t_{n+1}$.

When fluid particle i is in the interface region, the continuity equation remains the same as Eq.(46). On the other hand, the average values of density and viscosity are adopted in the

momentum equation, explained as follows. Interface implementation is a crucial step for the multi-phase fluid flow simulation. The discontinuities will present in the interface region. The non-local approximation near the discontinuous region will suffer numerical errors which will be more serious with higher-order schemes (Rezavand et al., 2018). As a result, in order to prevent numerical oscillations and unphysical particle penetrations (TingYe et al., 2019), the viscosity coefficient and the density for multi-phase interface interaction are smoothed in a harmonic mean interpolation (Adami et al., 2012; Pahar and Dhar, 2016). For example, if particle i and particle j belong to different fluid phases, interphase properties for density, ρ_{ij} and viscosity coefficient μ_{ij} are used (Adami et al., 2012; Pahar and Dhar, 2016). Therefore, the momentum equation is updated as:

$$\mathbf{a}_i^{n+1} = \frac{1}{\rho_i^{n+1}} \left(\begin{array}{l} -\sum_{j=1}^{N_i} (P_j^n - P_i^n) \mathbf{g}_1(\xi_{ij}^n) V_j + \sum_{j=1}^{N_i} \mu_{ij} (\mathbf{v}_j^n - \mathbf{v}_i^n) \text{Tr}(\mathbf{g}_2(\xi_{ij}^n)) V_j \\ -\beta \frac{\left(\sum_{j=1}^{N_i} \min(N_i, N_j) (\varphi \hat{\mathbf{n}}_j^n - \hat{\mathbf{n}}_i^n) \cdot \mathbf{g}_1(\xi_{ij}^n) V_j \right)}{\sum_{j=1}^{N_i} \min(N_i, N_j) w_0(\xi_{ij}^n) V_j / \sum_{j=1}^{N_i} w_0(\xi_{ij}^n) V_j} \left(\sum_{j=1}^{N_i} \mathbf{C}_{ij} \mathbf{g}_1(\xi_{ij}^n) V_j \right) \end{array} \right) + \frac{1}{\rho_i^{n+1}} \mathbf{F}^B \quad (48)$$

$$\mu_{ij} = \frac{2\mu_i\mu_j}{\mu_i + \mu_j} \quad (49a)$$

$$\rho_{ij} = \frac{2\rho_i\rho_j}{\rho_i + \rho_j} \quad (49b)$$

where μ_i and μ_j represent the viscosity coefficients of particle i and j . Similarly, ρ_i and ρ_j are the densities for particle i and j , respectively. It should be mentioned that if a high density ratio or a high viscosity ratio is involved, the simple averaging equation Eq.(49) overestimates the heavier or more viscous fluid effect on the other fluid. Besides, more numerical treatments are required to improve the stability of the current algorithm to simulate fluid flows at high Reynold numbers. For example, the novel methodology and numerical techniques in (Nangia et al., 2019a; Nangia et al., 2019b; Patel and Natarajan, 2017) such as a novel preconditioner used to simultaneously solve velocity and pressure can be incorporated. Therefore, more investigations are required to make the current numerical scheme being more accurate and

stable when high density and viscosity ratios are involved. In the present study, only low density ratio and low Reynold number multi-phase fluid flows are considered.

The velocity and displacements can be predicted by using the Euler forward method (1st order) or the Velocity Verlet algorithm (2nd order) (Sloper et al., 1982) as

Euler forward method:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{a}_i^{n+1} \Delta t \quad (50a)$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \mathbf{v}_i^{n+1} \Delta t \quad (50b)$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^0 + \mathbf{u}_i^{n+1} \quad (50c)$$

Velocity Verlet algorithm:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \frac{1}{2} (\mathbf{a}_i^{n+1} + \mathbf{a}_i^n) \Delta t \quad (51a)$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \mathbf{v}_i^n \Delta t + \frac{1}{2} \mathbf{a}_i^n \Delta t^2 \quad (51b)$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^0 + \mathbf{u}_i^{n+1} \quad (51c)$$

where the displacement of particle i is denoted by \mathbf{u}_i . The term \mathbf{x}_i^0 represents the initial coordinate for particle i , and \mathbf{x}_i^{n+1} represents the updated coordinate at $t = t_{n+1}$.

The time step size Δt should satisfy the CFL condition for kinetic, viscous, body force and surface tension force fields, shown as (Krimi et al., 2018)

$$\Delta t < \min \left(\frac{h}{|\mathbf{V}_{\max}| + c_B}, \frac{1}{8} \frac{h^2}{\max(\mu_A/\rho_{0,A}, \mu_B/\rho_{0,B})}, \frac{1}{4} \sqrt{\frac{h}{g}}, \frac{1}{4} \sqrt{\frac{\min(\rho_{0,A}, \rho_{0,B}) h^3}{2\pi\beta}} \right) \quad (52)$$

where h is the smoothing length, $|\mathbf{V}_{\max}|$ is the maximum fluid particle velocity, μ is the viscosity coefficient, ρ is the fluid density, g is the gravity acceleration, and β is the surface tension coefficient. The artificial speed of sound for the lighter fluid is larger than the one for

the heavier fluid on the condition of same material constants $\gamma_A = \gamma_B$. Therefore, the value of c_B is used for the constraints on the time step size.

4.2 Boundary Treatment

4.2.1 Boundary conditions for solid wall

In the present work, the fictitious particles are employed to represent the solid boundaries as described in (Gao and Oterkus, 2019b, c).

Non-slip boundary conditions

The velocities of the fictitious wall particles can be calculated based on the velocities of the fluid particles. As illustrated in Fig. 3, for the non-slip solid boundary condition, the velocity of the solid fictitious particle i , is calculated as (Adami et al., 2012)

$$\mathbf{v}_i^n = 2\mathbf{v}_{wall} - \left(\sum_{j=1}^{N_A} w_0(|\xi_{ij}^n|) \mathbf{v}_j^n + \sum_{k=1}^{N_B} w_0(|\xi_{ik}^n|) \mathbf{v}_k^n \right) \quad (53)$$

where N_A and N_B represent the number of family members of material point i that belongs to fluid A and B, respectively. Note that material point k and j cannot be fictitious wall particles. Please note that in Eq.(53) fictitious wall particles are not included in the family members of material point i .

Furthermore, particles j and k represent the family members of the solid particle i , where particle j belongs to fluid A and particle k belongs to fluid B. Their velocities in the current configuration are denoted by \mathbf{v}_j^n and \mathbf{v}_k^n , respectively. The parameter \mathbf{v}_{wall} denotes the prescribed solid wall velocity. The second term on the right-hand side represents the weighted fluid velocity summation. In Eq. (52), w_0 represents the improved Gaussian weighted function as provided in Eq. (40).

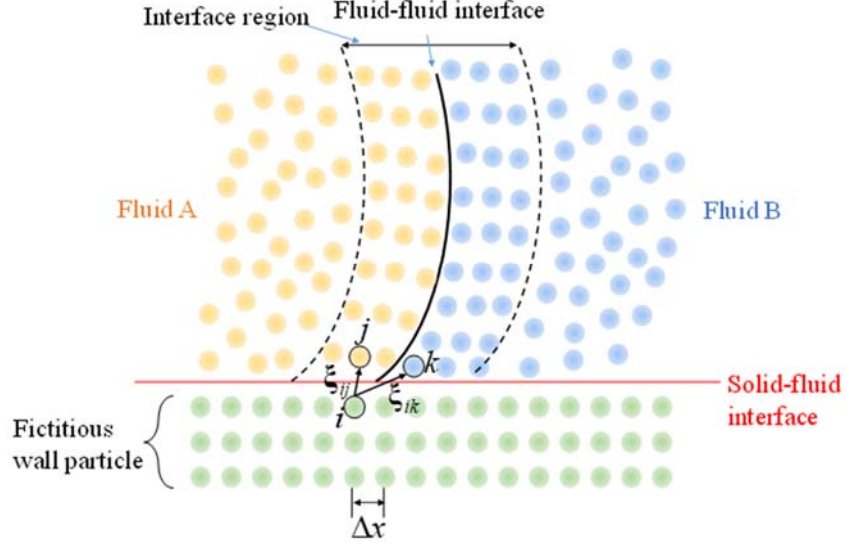


Fig. 3 Boundary and interface illustration

The pressure of the solid fictitious particle is calculated based on the fluid particles according to the formulation provided in (Adami et al., 2012) as

$$P_i = \frac{\sum_{j=1}^{N_A} (P_j + (\mathbf{F}_j^B - \rho_j \mathbf{a}_{wall}) \cdot \boldsymbol{\xi}_{ij}^n) w_0(|\boldsymbol{\xi}_{ij}^n|) + \sum_{k=1}^{N_B} (P_k + (\mathbf{F}_k^B - \rho_k \mathbf{a}_{wall}) \cdot \boldsymbol{\xi}_{ik}^n) w_0(|\boldsymbol{\xi}_{ik}^n|)}{\sum_{j=1}^{N_A} w_0(|\boldsymbol{\xi}_{ij}^n|) + \sum_{k=1}^{N_B} w_0(|\boldsymbol{\xi}_{ik}^n|)} \quad (54a)$$

$$\mathbf{F}_j^B = \rho_j \mathbf{g}; \mathbf{F}_k^B = \rho_k \mathbf{g}; \quad (54b)$$

$$\rho_i = \rho_{0,i} \left(\frac{P_i - P_{0,i}}{P_{ref,i}} + 1 \right)^{1/\gamma_i} \quad (54c)$$

where P_j and P_k represent the fluid pressure. \mathbf{F}_j^B and \mathbf{F}_k^B represent the body force of the fluid particles. In the current study, the body force is calculated as Eq.(53b) with \mathbf{g} representing the gravitational acceleration. \mathbf{a}_{wall} represents the acceleration of the solid wall. $\rho_{0,i}$, $P_{0,i}$, $P_{ref,i}$ and γ_i are the initial density, background pressure, reference pressure, and material constant for particle i . Please note that in Eq.(53a) fictitious wall particles are not included in the family members of material point i .

Slip boundary condition for solid wall

For the slip boundary condition, the viscous forces are neglected for fluid-solid particle interactions ($\mathbf{F}^v = 0$) and the velocity of the fictitious wall particle is set as (Adami et al., 2012)

$$\mathbf{v}_i^n = \mathbf{v}_{wall} \quad (55)$$

The pressure of the fictitious solid particle is calculated according to Eq.(54a) .

4.2.2 Boundary collision model

If the fluid particle penetrates the solid wall in the updated configuration, the velocity of the fluid particle needs to be corrected (Ren and Zhuang, 2018). In this case, the boundary collision model (Ren and Zhuang, 2018) can be applied where the solid wall is simulated as a reflective boundary condition as

$$\left(\mathbf{v}_i^{n+1}\right)^{corrected} = \left(\mathbf{v}_i^{n+1}\right) - 2\left(\left(\mathbf{v}_i^{n+1} - \mathbf{v}_{wall}\right) \cdot \hat{\mathbf{n}}_{wall}\right) \hat{\mathbf{n}}_{wall}, \text{ if } \left(\mathbf{v}_i^{n+1} - \mathbf{v}_{wall}\right) \cdot \hat{\mathbf{n}}_{wall} < 0 \quad (56)$$

where $\hat{\mathbf{n}}_{wall}$ represents the unit normal direction of the solid wall, as shown in Fig. 4.

The velocity correction in Eq.(56) is illustrated in Fig. 4.

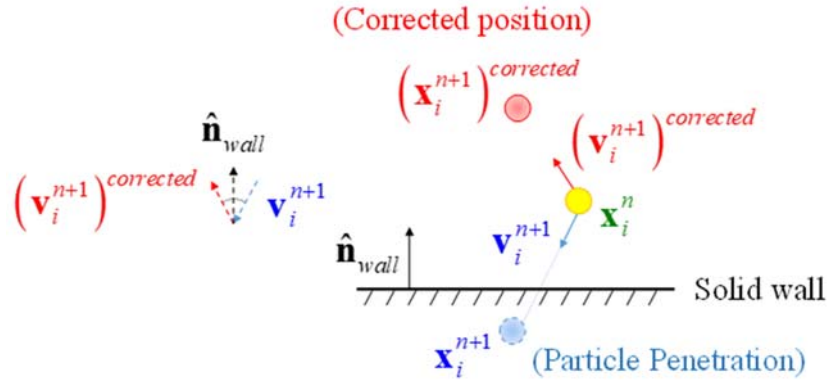


Fig. 4 Sketch for solid wall collision model

4.3 Particle shifting technology for preventing particle clustering

For particle methods such as SPH, the instability problem may occur when the particle distribution becomes highly distorted (Akbari, 2019). The numerical errors induced by the irregular particle spacing may dominate the results especially in the later stage of the numerical

simulations. The same problem also happens in the PD field, as explained by Silling et al. (Silling et al., 2017). To avoid the particle clustering phenomenon or the instability problem, the particle shifting technology (PST) (Xu et al., 2009) is widely applied in the fluid flow simulations (Akbari, 2019; Lind et al., 2012; Mokos et al., 2017; Shadloo et al., 2012; Vacondio et al., 2013; Zhang et al., 2019). The particles will be nearly uniformly distributed after shifting; thus, the singularity of the particle distribution will be reduced. Therefore the particle positions can be modified by using a shifting displacement (Xu et al., 2009):

$$\left(\mathbf{x}_i^{n+1}\right)^{corrected} = \mathbf{x}_i^0 + \left(\mathbf{u}_i^{n+1}\right)^{shifted} \quad (57)$$

The shifting displacement for particle i can be found as

$$\left(\mathbf{u}_i^{n+1}\right)^{shifted} = \mathbf{u}_i^{n+1} + \left(\delta\mathbf{u}_i\right)^{PST} \quad (58)$$

In Eq. (58), the parameter $\left(\delta\mathbf{u}_i\right)^{PST}$ represents the shifting distance and it can be defined as

$$\left(\delta\mathbf{u}_i\right)^{PST} = C_{PST}\alpha_{PST}\mathbf{U}_i \quad (59)$$

where C_{PST} is a predefined constant ranging from 0.01 to 0.1 depending on the specific problem and α_{PST} is the shifting magnitude which can be defined as (Xu et al., 2009)

$$\alpha_{PST} = \left|\mathbf{V}_{\max}\right|\Delta t \quad (60)$$

where $\left|\mathbf{V}_{\max}\right|$ can be chosen as the magnitude of the largest velocity overall fluid particles or it can be estimated specifically according to the specific problem (Xu et al., 2009) .

In Eq. (59), the parameter \mathbf{U}_i represents the displacement shift-vector which can be defined as (Xu et al., 2009)

$$\mathbf{U}_i = \sum_{j=1}^{N_i} \frac{\left|\bar{\xi}_i\right|^2}{\left|\xi_{ij}^n\right|^2} \zeta_{ij} \quad (61a)$$

with

$$\zeta_{ij} = \frac{\xi_{ij}^n}{|\xi_{ij}^n|} \quad (61b)$$

$$|\bar{\zeta}_i| = \frac{1}{N_i} \sum_{j=1}^{N_i} |\xi_{ij}^n| \quad (61c)$$

where $|\bar{\zeta}_i|$ represents the average spacing between the central point i and its family members (e.g. j) (Xu et al., 2009) and ζ_{ij} represents the unit distance vector between particles i and j . Note that, the summation of ζ_{ij} in Eq. (61a) can reflect the anisotropy of the current particle distribution (Xu et al., 2009) and the term $\left(\frac{|\bar{\zeta}_i|^2}{|\xi_{ij}^n|^2} \right)$ acts as a weighted function in the formulation. According to (Lind et al., 2012), numerical errors will be induced if the shifting distance is much larger than the initial spacing distance Δx . Therefore, an upper limit for the particle shifting distance, $(\delta \mathbf{u}_i)^{PST}$ is imposed as $0.2\Delta x$ (Lind et al., 2012). As an optional procedure, the velocity and pressure can also be corrected based on the displacement shifting. The corresponding details are provided in Appendix B. The flowchart of the PST is provided in Fig. 5 as

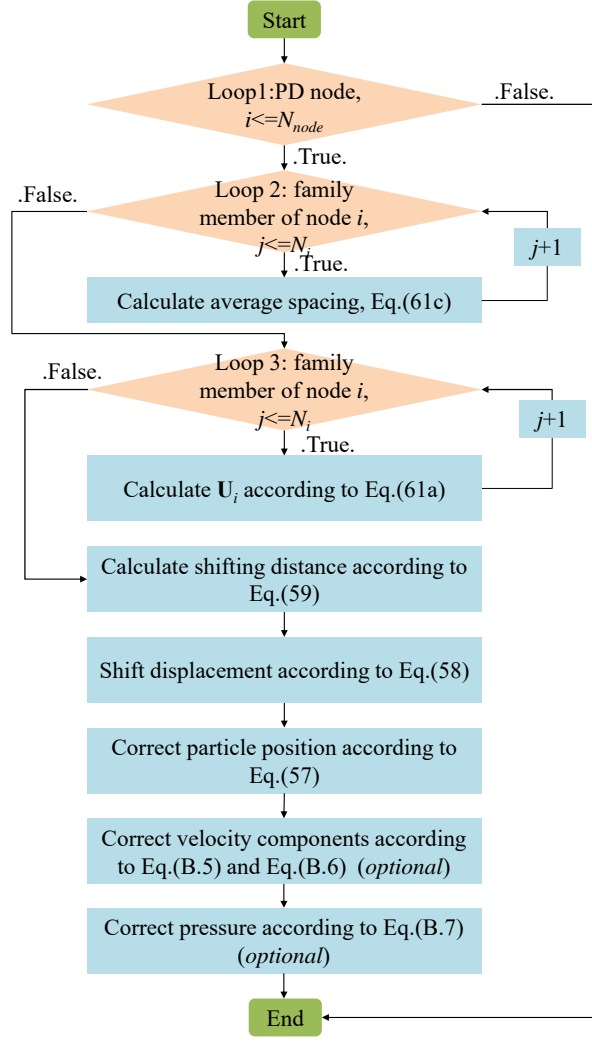


Fig. 5 Flowchart for PST algorithm

4.4 Moving least square algorithm for preventing pressure instability

It is known that the free movements of particles may create the “pressure instability” problems for Lagrangian particle methods, where the mass is not conserved and great oscillations occur in density and pressure fields (Chen et al., 2013; Chen et al., 2015). In order to overcome this problem, velocity, pressure and density values are smoothed by using moving least square (MLS) algorithm (Dilts, 1999, 2000). The algorithm is introduced to hydrodynamics by Dilts (Dilts, 1999). In the present work, the first order MLS weighted function is calculated according to (Colagrossi and Landrini, 2003) as

$$w_{MLS}(\xi_{ij}^n) = \left[\beta_0(\mathbf{x}_i^n) + \beta_1(\mathbf{x}_i^n)(-\xi_{ij}^1) + \beta_2(\mathbf{x}_i^n)(-\xi_{ij}^2) \right] w_0(\xi_{ij}^n) \quad (62)$$

with

$$\xi_{ij}^n = \mathbf{x}_j^n - \mathbf{x}_i^n = \begin{bmatrix} \xi_{ij}^1 \\ \xi_{ij}^2 \end{bmatrix} \quad (63a)$$

$$\boldsymbol{\beta}(\mathbf{x}_i^n) = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} = \mathbf{S}^{-1}(\mathbf{x}_i^n) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (63b)$$

$$\mathbf{S}(\mathbf{x}_i^n) = \sum_{j=1}^{N_i} w_0(\xi_{ij}^n) \begin{bmatrix} -1 & -\xi_{ij}^1 & -\xi_{ij}^2 \\ -\xi_{ij}^1 & (\xi_{ij}^1)^2 & \xi_{ij}^1 \xi_{ij}^2 \\ -\xi_{ij}^2 & \xi_{ij}^1 \xi_{ij}^2 & (\xi_{ij}^2)^2 \end{bmatrix} \quad (63c)$$

where \mathbf{x}_i^n and \mathbf{x}_j^n are the positions of particle i and j in the current configuration. It should be pointed out that the weighted function $w_0(\xi_{ij}^n)$ adopted in MLS is the improved Gaussian function (Colagrossi and Landrini, 2003) provided in Eq.(40) which is different from the Gaussian function in Eq. (7) used in the PDDO construction process. In MLS algorithm, the smoothing length h in the improved Gaussian function Eq. (40) is set as $1.2\Delta x$. Besides, the horizon size is set to $\delta = 3h$. As a result, if the MLS algorithm is employed, the horizon size is set as $\delta = 3.6\Delta x$, in order to have the same family members of a particle both in PDDO calculation and MLS algorithm. To lower the computation cost, the MLS scheme is adopted every n_{MLS} time steps (Chen et al., 2015).

Next, the velocity and pressure can be smoothed as

$$(\mathbf{v}_i^n)^{smoothed} = \frac{\sum_j (\mathbf{v}_j^n) w_{MLS}(\xi_{ij}^n) V_j}{\sum_j w_{MLS}(\xi_{ij}^n) V_j} \quad (64a)$$

$$(P_i^{n+1})^{smoothed} = \frac{\sum_j (P_j^n) w_{MLS}(\xi_{ij}^n) V_j}{\sum_j w_{MLS}(\xi_{ij}^n) V_j} \quad (64b)$$

Besides, the density is smoothed as (Chen et al., 2015)

$$(\rho_i^n)^{smoothed} = \frac{\sum_j (\tilde{\rho}_j^n) w_{MLS}(\xi_{ij}^n) V_j}{\sum_j w_{MLS}(\xi_{ij}^n) V_j} \quad (65)$$

with

$$(\tilde{\rho}_j^n) = \left(\frac{(P_j^n) - P_0}{P_{ref}} + 1 \right)^{\frac{1}{\gamma_i}} \rho_{0,i} \quad \text{and} \quad \gamma_i = \begin{cases} \gamma_A & \text{for fluid A} \\ \gamma_B & \text{for fluid B} \end{cases}; \quad \rho_{0,i} = \begin{cases} \rho_{0,A} & \text{for fluid A} \\ \rho_{0,B} & \text{for fluid B} \end{cases} \quad (66)$$

where $w_{MLS}(\xi_{ij}^n)$ is provided in Eq.(62). The parameter $\rho_{0,i}$ is the initial density of particle i . The correction in Eq. (66) can be considered as converting the phase of particle j into the phase of particle i (Chen et al., 2015). The flowchart for the MLS implementation is provided in Fig. 6 as (Gao and Oterkus, 2019c)

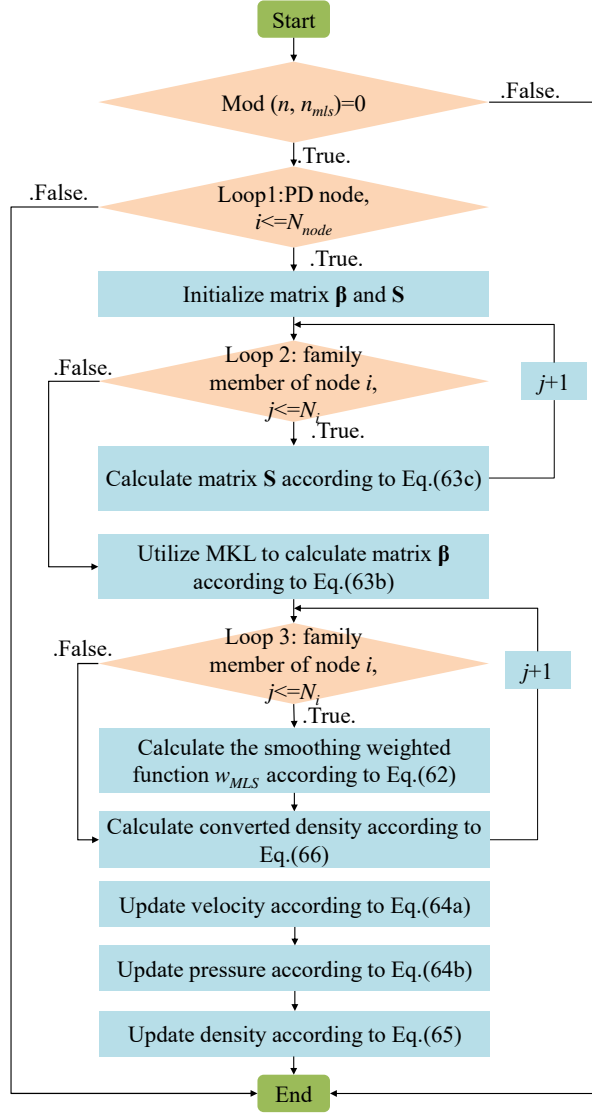


Fig. 6 Flowchart for MLS algorithm

4.5 XSPH displacement smoothing

Compared to the PST introduced in Section 4.3 for large irregular particle distributions, a simpler XSPH method can be chosen for less irregular particle distributions to save computational time. The formulation in XSPH method (Monaghan, 1992) is adopted here to only update the displacements as

$$\left(\mathbf{u}_i^{n+1}\right)^{shifted} = \mathbf{u}_i^{n+1} + \left(\delta \mathbf{u}_i\right)^{XSPH} \quad (67a)$$

with

$$\left(\delta \mathbf{u}_i\right)^{XSPH} = \left(\chi \sum_j m_j \frac{\left(\rho_i^n + \rho_j^n\right)}{2 \rho_i^n \rho_j^n} \left(\mathbf{v}_j^n - \mathbf{v}_i^n\right) w_0\left(\left|\xi_{ij}^n\right|\right) \right) \Delta t \quad (67b)$$

where χ is an adjustable parameter as $0 \leq \chi \leq 1$ (Monaghan, 1992) and m_i is the mass of particle i which can be calculated as

$$m_i = \rho_{0,i} V_i \quad (67c)$$

4.6 Artificial viscosity

To reduce the undesirable oscillations, the artificial viscosity, Π , widely applied in SPH method (Monaghan, 1992) can be utilized and incorporated into the current PDDO model. According to (Monaghan, 1992; Monaghan and Gingold, 1983), the momentum equation Eq.(13) can be modified by including the artificial viscosity form as

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla P + \mathbf{F}^V + \mathbf{F}^B + \mathbf{F}^S + \nabla \Pi \quad (68)$$

where the SPH form of the gradient of artificial viscosity Π_i for particle i is expressed as (Suresh et al., 2019b)

$$\nabla \Pi_i = \begin{cases} \sum_{j=1}^{N_i} \frac{\alpha h c_{avg} (\mathbf{v}_j - \mathbf{v}_i) \cdot \boldsymbol{\xi}_{ij}}{|\boldsymbol{\xi}_{ij}|^2 + (0.1h)^2} (\nabla_i W_{ij}) V_j, & \text{for } (\mathbf{v}_j - \mathbf{v}_i) \cdot \boldsymbol{\xi}_{ij} < 0 \\ 0, & \text{otherwise} \end{cases} \quad (69)$$

where W_{ij} represents a weighted function in SPH method.

In Eq.(69), c_{avg} is the average value of the artificial speeds of sound of particle i and j . Inspired by SPH method (Monaghan, 1992), the gradient of the artificial viscosity term Π_i is reformulated by replacing the SPH gradient operator with PDDO as

$$\nabla \Pi_i = \begin{cases} \sum_{j=1}^{N_i} \frac{\alpha h c_{avg} (\mathbf{v}_j^n - \mathbf{v}_i^n) \cdot \boldsymbol{\xi}_{ij}^n}{|\boldsymbol{\xi}_{ij}^n|^2 + (0.1h)^2} (\mathbf{g}_1(\boldsymbol{\xi}_{ij}^n)) V_j, & \text{for } (\mathbf{v}_j^n - \mathbf{v}_i^n) \cdot \boldsymbol{\xi}_{ij}^n < 0 \\ 0, & \text{otherwise} \end{cases} \quad (70)$$

where $h = 1.2\Delta x$ (being same with the one in MLS scheme) (Liu et al., 2019) and the empirical constant α which generally ranges from 0.01 to 0.1 (Monaghan, 1992). In this study, the empirical constant is chosen as $\alpha = 0.02$.

Finally, the flowchart of the algorithm including surface tension calculation is provided in Fig. 7 (Gao and Oterkus, 2019c). Sequential computing is used in the present study. The horizon size and the shape of the neighbourhood are fixed in the time integration. Therefore, the family members of one PD point are updated every time step.

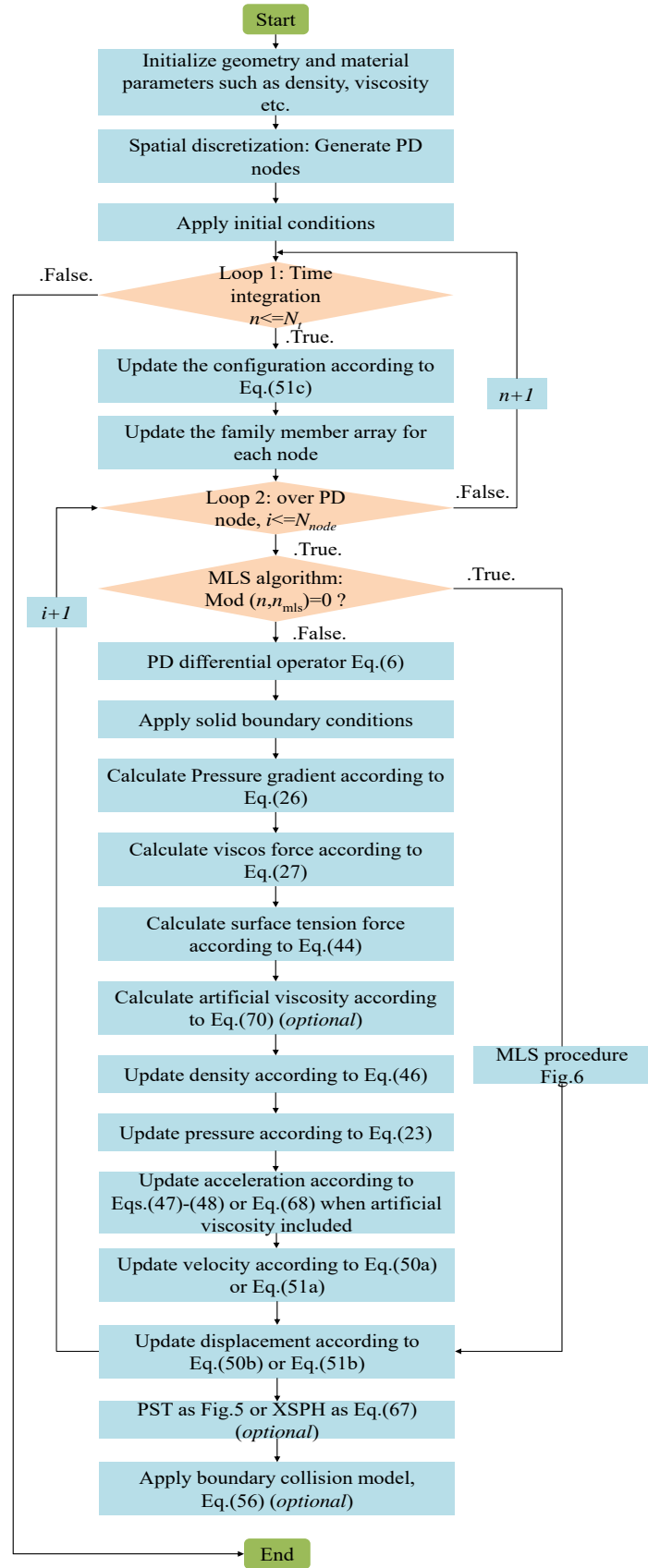


Fig. 7 Flowchart for the multi-phase fluid flow simulation by the PDDO model including the surface tension force calculation

5 Numerical Simulations

In this section, some multi-phase fluid flow benchmark problems are solved by using the newly developed PDDO model. First, the numerical accuracy of the PPDO is tested for both function and derivative approximations. Second, the two-phase fluids hydrostatic problem is simulated to validate the energy conservation of the PDDO model. Third, the two-phase Poiseuille flow simulation is conducted to validate the multi-viscosity coefficient treatment. Subsequently, the 2D droplet deformation under the surface tension force is simulated to validate the surface tension PDDO formulation.

5.1 PPDO model accuracy test

5.1.1 Uniform and non-uniform particle distribution effect on PDDO accuracy

Prior to the multi-phase fluid flow simulations, the PDDO accuracy is tested for first and second order derivatives both for uniform and non-uniform particle distributions. In this way, the effect of the non-uniform particle distribution which frequently exists in fluid flow simulation on the PDDO accuracy is investigated.

In order to test the higher-order derivatives approximation accuracy, a function is set as an object

$$f(x, y) = x^2 + 2y^3 \quad (71)$$

with the variable domain of $x \in [0, 2]$ and $y \in [0, 2]$. The exact solutions for the derivatives are easy to obtain as

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &= 2x; \quad \frac{\partial f(x, y)}{\partial y} = 6y^2; \\ \frac{\partial^2 f(x, y)}{\partial x^2} &= 2; \quad \frac{\partial^2 f(x, y)}{\partial y^2} = 12y; \quad \frac{\partial^2 f(x, y)}{\partial x \partial y} = 0 \end{aligned} \quad (72)$$

In the numerical implementation, the domain is discretized by 21×21 particles with both uniform and non-uniform distributions, as shown in Fig. 8. The non-uniform distribution is generated from a standard normal distribution (Adams et al., 1997). The red point located at (1,1) in Fig. 8 is set to be the point of interest.

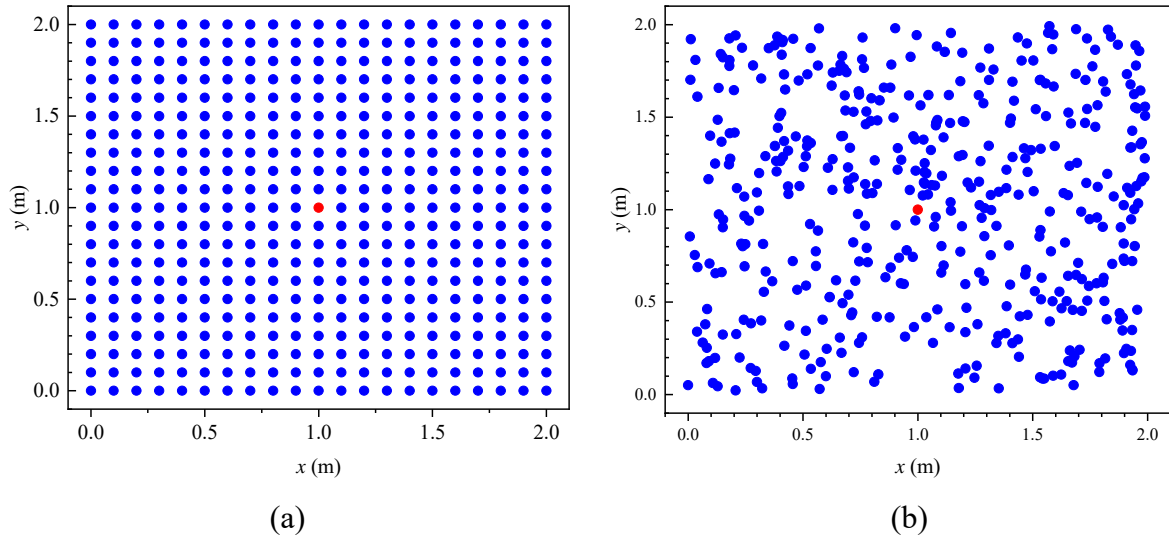


Fig. 8 (a) Uniform (b) non-uniform particle distributions
 The values of derivatives obtained by PD in uniform particle distribution, PD in non-uniform particle distribution, and analytical solution are provided in Fig. 9-Fig. 13.

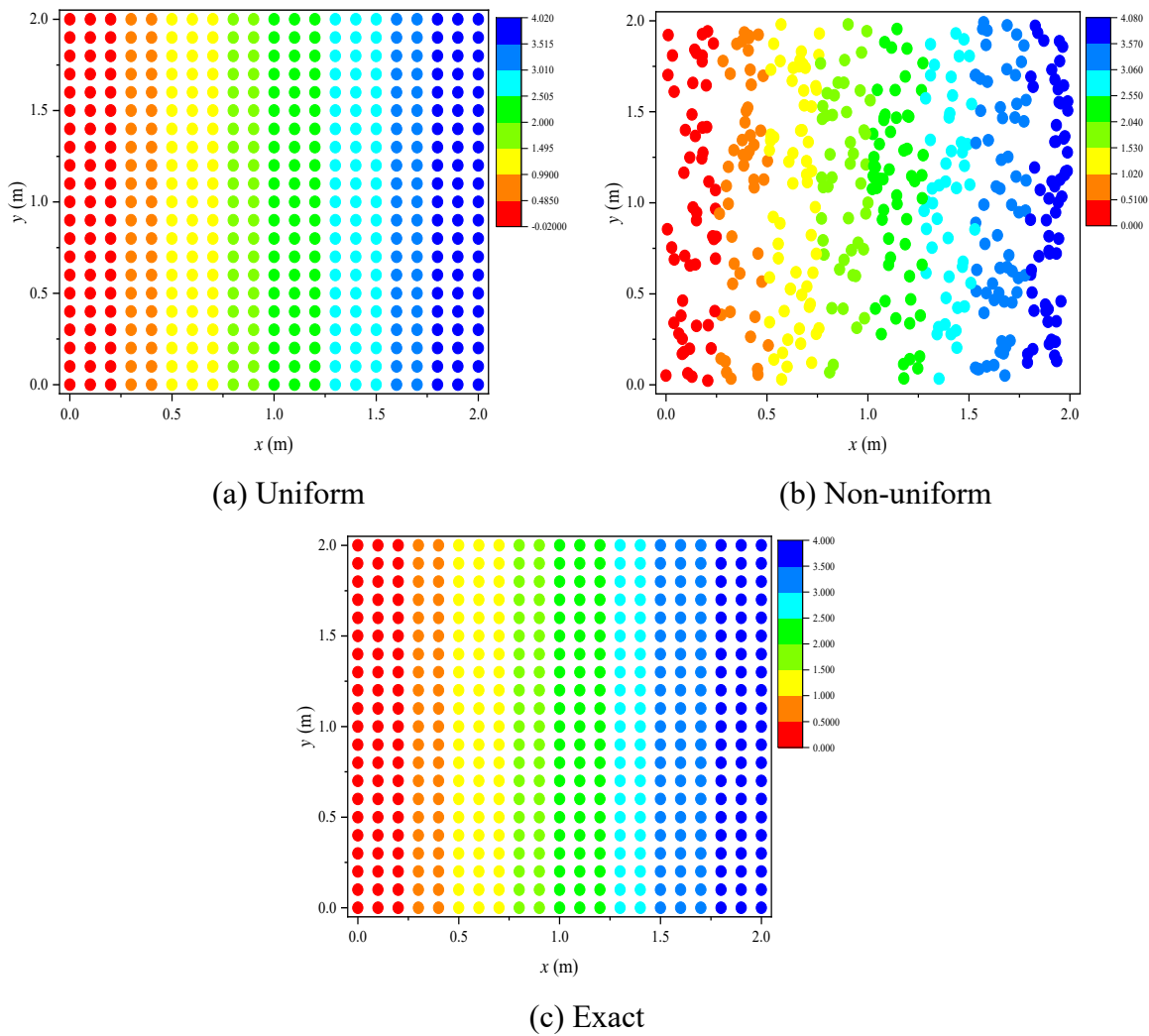


Fig. 9 (a) Uniform (b) non-uniform particle distributions, and (c) exact values for $\partial f / \partial x$

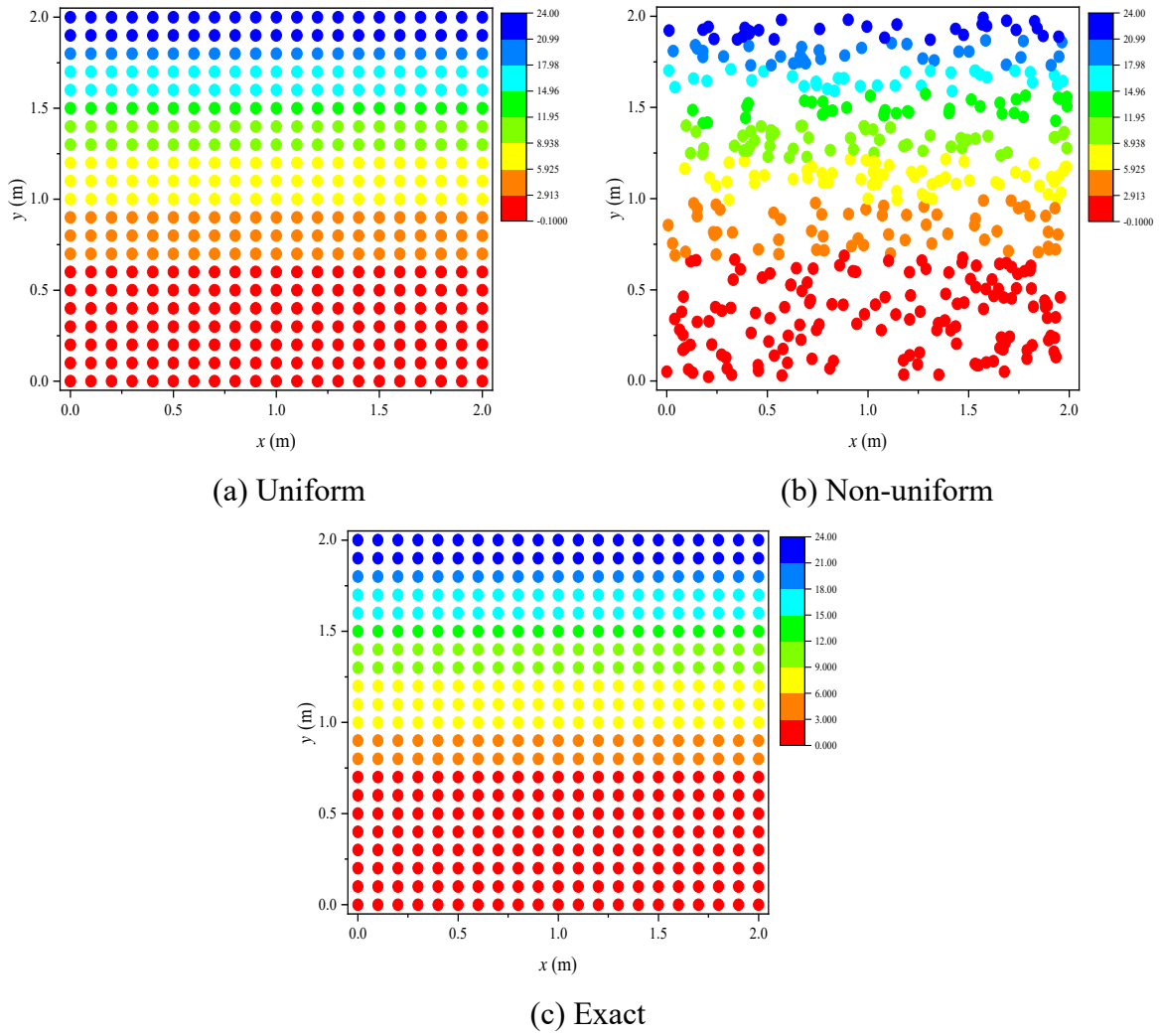


Fig. 10 (a) Uniform (b) non-uniform particle distributions, and (c) exact values for $\partial f / \partial y$

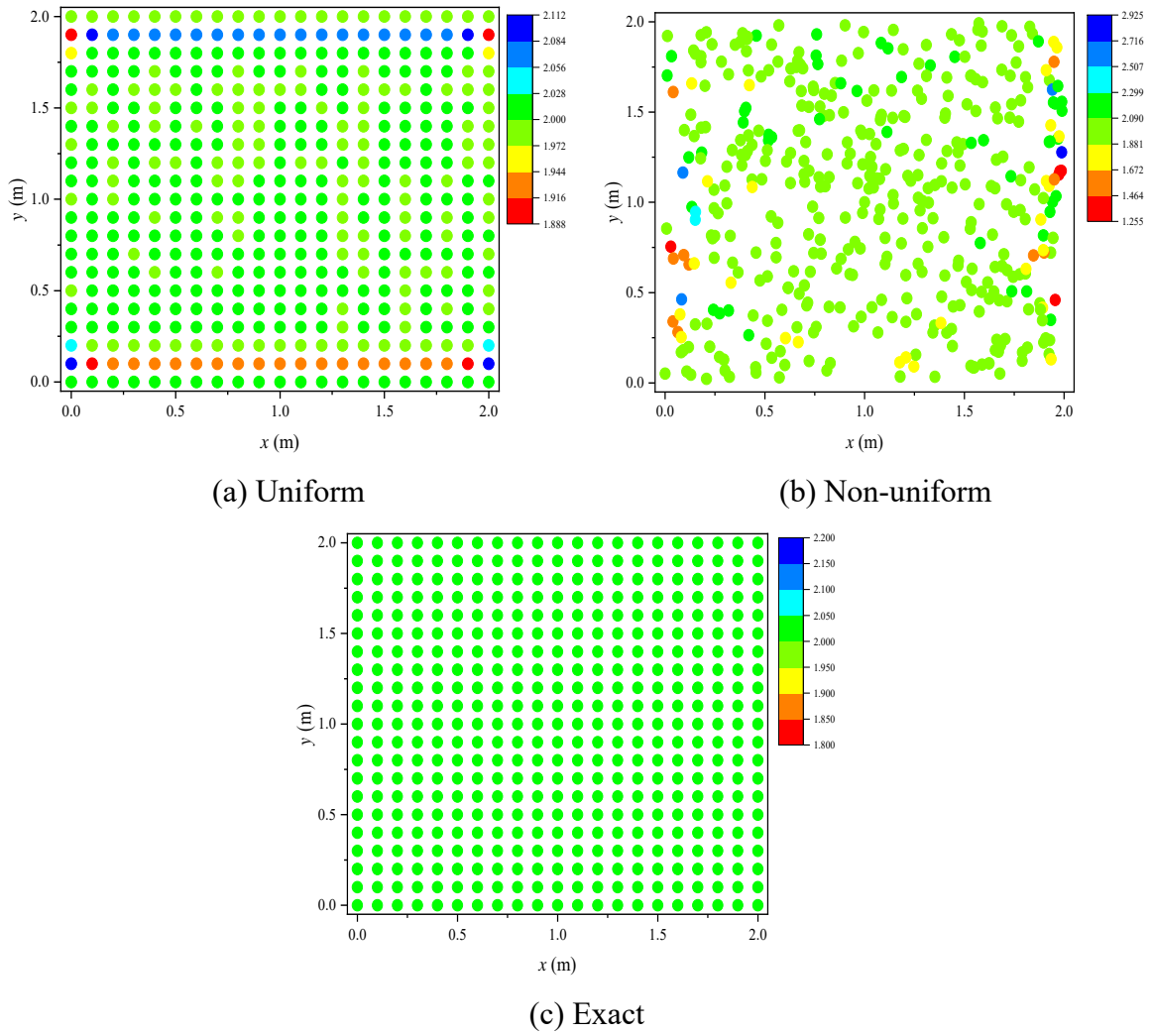


Fig. 11 (a) Uniform (b) non-uniform particle distributions, and (c) exact values for $\partial^2 f / \partial x^2$

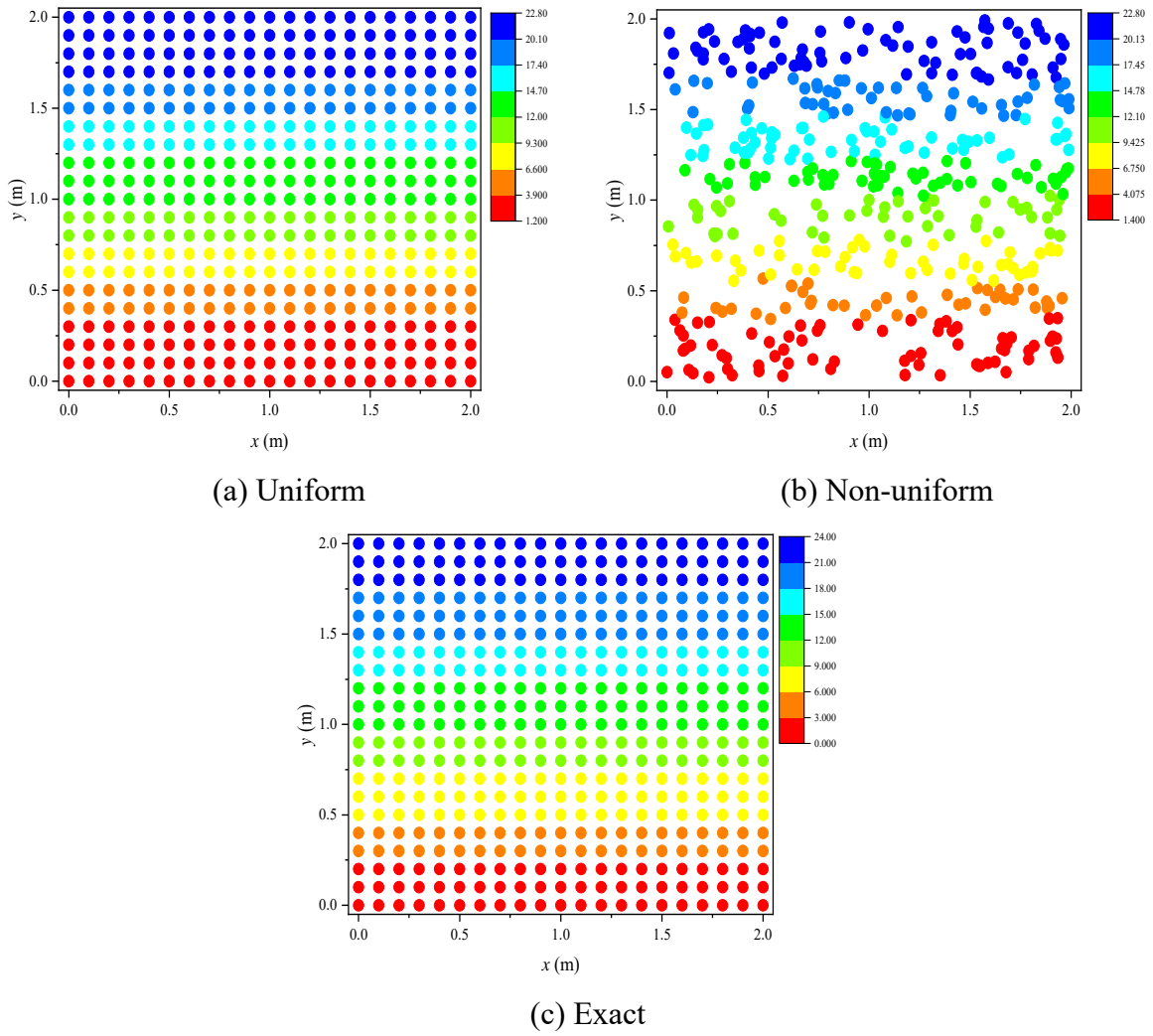


Fig. 12 (a) Uniform (b) non-uniform particle distributions, and (c) exact values for $\partial^2 f / \partial y^2$

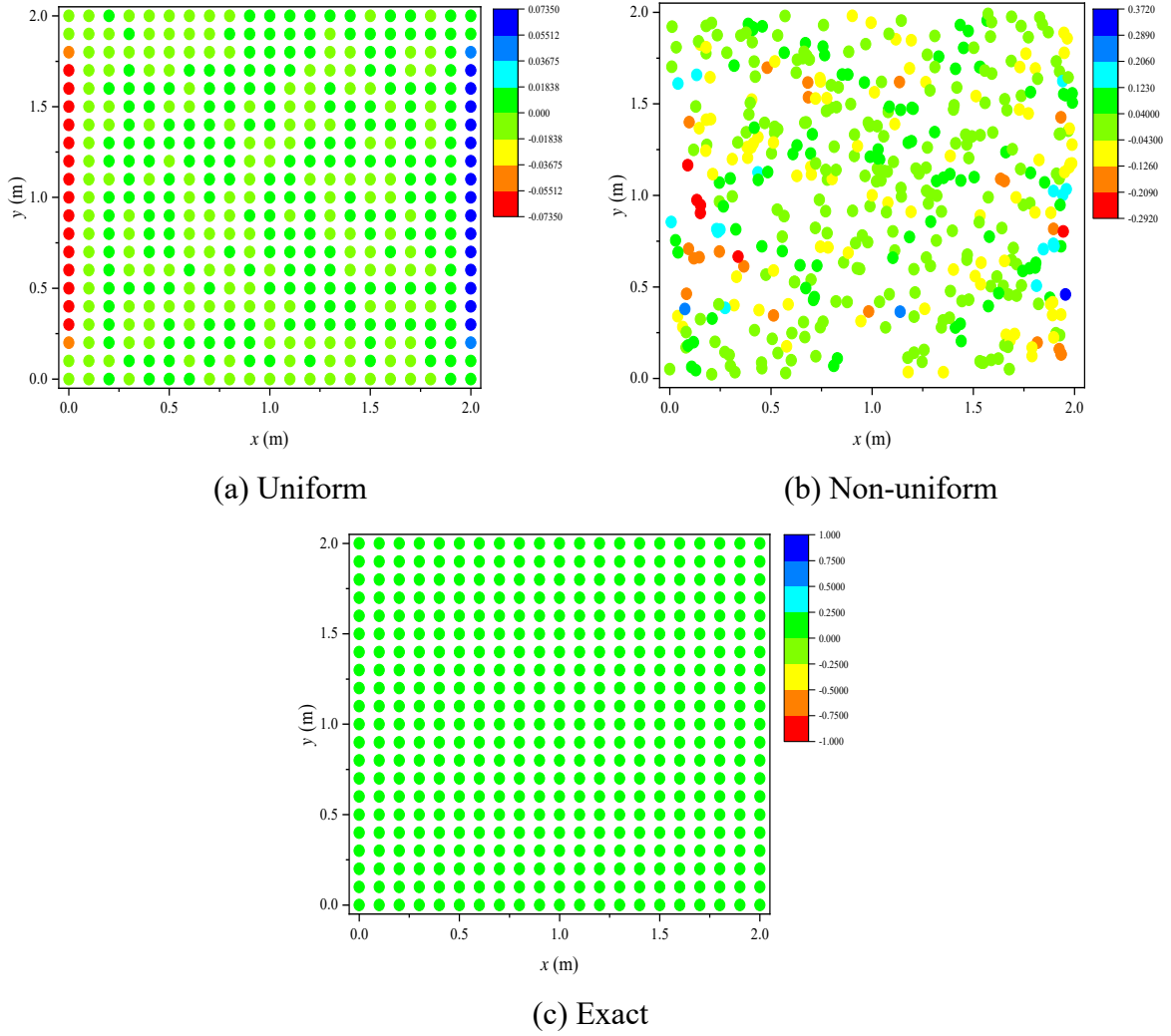


Fig. 13 (a) Uniform (b) non-uniform particle distributions, and (c) exact values for $\partial^2 f / \partial x \partial y$

The average relative error between the PD predicted value and exact value is calculated for all the derivatives with PD nodes 21×21 , 51×51 , 81×81 , 101×101 . The differences between PD values and exact values for derivatives f_x , f_y , f_{xx} and f_{yy} are compared by using average relative error. The average relative error for f_{xx} is calculated as

$$\mathcal{E}_r = \left(\sum_{i=1}^N \left| \frac{f_{xx}^{PD}(i) - f_{xx}^{Exact}(i)}{f_{xx}^{Exact}(i)} \right| / N \right) \times 100\% \quad (73a)$$

Since the exact value of $\partial^2 f / \partial x \partial y$ is zero, the average relative error for $\partial^2 f / \partial x \partial y$ is calculated as

$$\varepsilon_r = \left(\sum_{i=1}^N |f_{xy}^{PD}(i) - f_{xy}^{Exact}(i)| / N \right) \times 100\% \quad (73b)$$

The error for derivatives at different resolutions at red point located at (1,1) in Fig. 8 are presented in Fig. 14. Correspondingly, the PD derivatives for 21×21 and 51×51 resolutions are also added in Table 1.

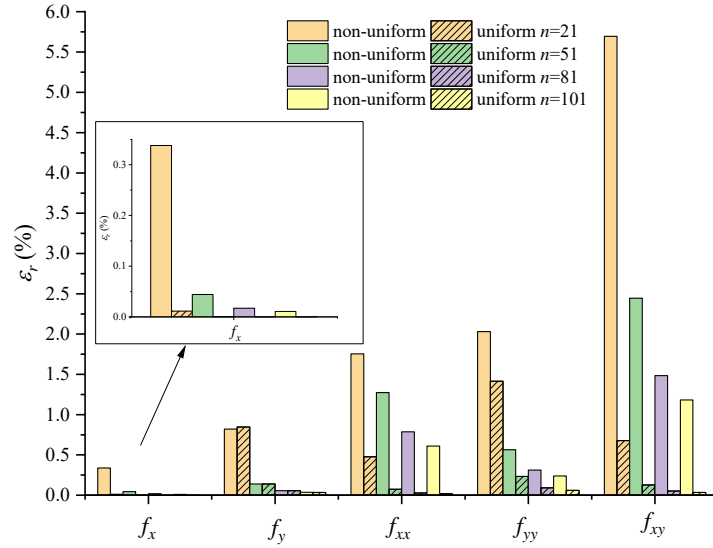


Fig. 14 L1-norm error for the derivatives at different resolutions

Table 1 Comparison of derivatives at $(x, y) = (1,1)$ with exact value and PDDO solution with 21×21 and 51×51 particles

Method		Exact Solution	PDDO Uniform		PPDO Non-uniform	
			value	error	value	error
$\partial f / \partial x$	21×21	2	2	0	2.0006	0.03%
	51×51		2	0	2.0001	0.005%
$\partial f / \partial y$	21×21	6	6.0536	0.89%	6.0572	0.95%
	51×51		6.0085	0.14%	6.008	0.13%
$\partial^2 f / \partial x^2$	21×21	2	2	0	1.9738	1.31%
	51×51		2	0	2.0034	0.68%
$\partial^2 f / \partial^2 y$	21×21	12	12	0	11.9924	0.063%
	51×51		12	0	12.0027	0.023%
$\partial^2 f / \partial x \partial y$	21×21	0	3.719E-15	-	-6.673E-2	-
	51×51		9.358E-16	-	6.531E-3	-

It can be observed from the above table that the accuracy of the PDDO both for first and second-order derivatives is acceptable. However, the irregularity of the particle distribution will increase the numerical error compared to the regular particle distribution as shown in Fig. 9-Fig. 13. Therefore, the techniques that redistribute the particles in a nearly uniform fashion is necessary for the PDDO simulations.

5.1.2 PDDO prediction accuracy test on PST corrections

The PST explained in Section 4.3 can be treated as a re-meshing method. The particles are redistributed from a disordered distribution to a nearly uniform one. Consequently, the velocity and pressure can be corrected on the new shifted position. In other words, although the particle position is shifted, the distribution of velocity and pressure should keep the same. The correction procedure of velocity and pressure are provided in Appendix B. The velocity or pressure can be generally treated as a function. Therefore, the procedure becomes predicting

the function value of a point on a new position from the values in old positions. The PDDO accuracy of this prediction procedure is tested in this subsection.

The non-uniform particles as shown in Fig. 8 (b) are treated as the particles before PST. The uniform distributed particles as shown in Fig. 8 (a) are set as the particles after PST. The function Eq.(71) is given to the non-uniform particles in Fig. 8 (b). The distribution of the function values $f(x, y)$ of these non-uniform particles is shown in Fig. 17 (a). The function values of the uniform particles (after PST) should be predicted by using the derivatives on the non-uniform particle positions calculated by PDDO. The procedure is provided in Fig. 15.

The positions of the uniform and non-uniform distributed particles are provided in Fig. 16 (a). The uniform distributed particles are set as the interest points with their family members being the non-uniform distributed points, as illustrated in Fig. 16 (b). Particle i represents the particle of interest belonging to the uniform distribution system, and particle j represents the particle belonging to the non-uniform distribution system within the horizon of particle i . Besides, the particles belonging to the uniform distribution are not the family members of particle i , although they are located within the horizon.

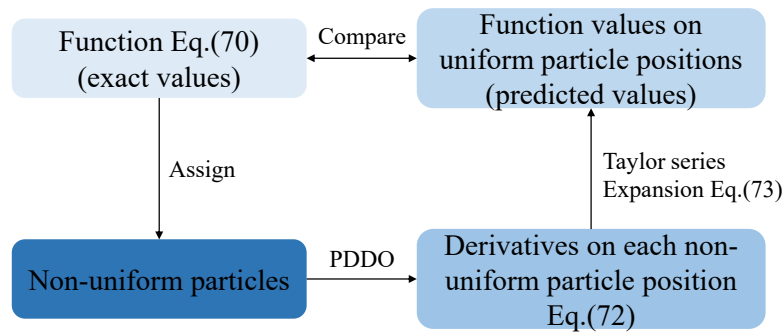
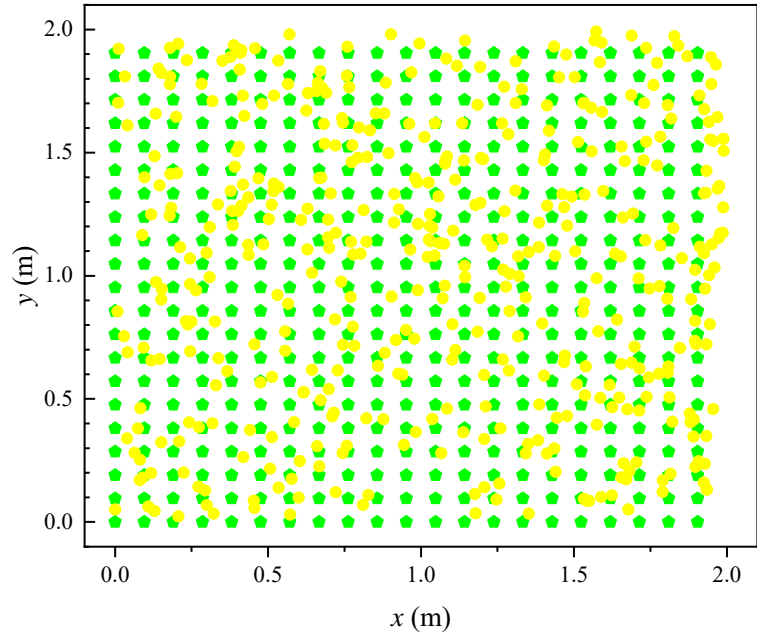
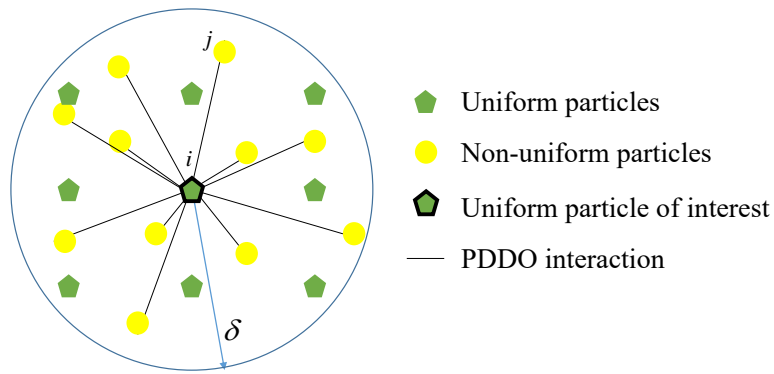


Fig. 15 Function value prediction procedure



(a)



(b)

Fig. 16 (a) Uniform and non-uniform distributed particle positions, (b) illustration of PDDO interaction between uniform particles and non-uniform particles

First, the derivatives of function $f(x, y)$ on the positions of non-uniform distributed particles are predicted by PDDO in the realm of the non-uniform distribution particle system. For example, for particle j belonging to the non-uniform particle distribution system, the derivatives are calculated according to Eq.(1) as

$$\begin{bmatrix} f_{,x}^{PDDO}(x_j^{nu}, y_j^{nu}) \\ f_{,y}^{PDDO}(x_j^{nu}, y_j^{nu}) \\ f_{,xx}^{PDDO}(x_j^{nu}, y_j^{nu}) \\ f_{,yy}^{PDDO}(x_j^{nu}, y_j^{nu}) \\ f_{,xy}^{PDDO}(x_j^{nu}, y_j^{nu}) \end{bmatrix} = \sum_{k=1}^{N_j} \left(f(x_k^{nu}, y_k^{nu}) - f(x_j^{nu}, y_j^{nu}) \right) \begin{bmatrix} g^{10}(\xi_{jk}) \\ g^{01}(\xi_{jk}) \\ g^{20}(\xi_{jk}) \\ g^{02}(\xi_{jk}) \\ g^{11}(\xi_{jk}) \end{bmatrix} V_k \quad (74a)$$

with

$$\begin{bmatrix} g^{10}(\xi_{jk}) \\ g^{01}(\xi_{jk}) \\ g^{20}(\xi_{jk}) \\ g^{02}(\xi_{jk}) \\ g^{11}(\xi_{jk}) \end{bmatrix} = \left(A^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)^T \begin{bmatrix} w(\xi_{jk}^1) \\ w(\xi_{jk}^2) \\ w(\xi_{jk}^1)^2 \\ w(\xi_{jk}^2)^2 \\ w(\xi_{jk}^1)(\xi_{jk}^2) \end{bmatrix} \quad (74b)$$

$$A = \sum_{k=1}^{N_j} w \begin{bmatrix} (\xi_{jk}^1)^2 & (\xi_{jk}^1)(\xi_{jk}^2) & (\xi_{jk}^1)^3 & (\xi_{jk}^1)(\xi_{jk}^2)^2 & (\xi_{jk}^1)^2(\xi_{jk}^2) \\ (\xi_{jk}^1)(\xi_{jk}^2) & (\xi_{jk}^2)^2 & (\xi_{jk}^1)^2(\xi_{jk}^2) & (\xi_{jk}^2)^3 & (\xi_{jk}^1)(\xi_{jk}^2)^2 \\ (\xi_{jk}^1)^3 & (\xi_{jk}^1)^2(\xi_{jk}^2) & (\xi_{jk}^1)^4 & (\xi_{jk}^1)^2(\xi_{jk}^2)^2 & (\xi_{jk}^1)^3(\xi_{jk}^2) \\ (\xi_{jk}^1)(\xi_{jk}^2)^2 & (\xi_{jk}^2)^3 & (\xi_{jk}^1)^2(\xi_{jk}^2)^2 & (\xi_{jk}^2)^4 & (\xi_{jk}^1)(\xi_{jk}^2)^3 \\ (\xi_{jk}^1)^2(\xi_{jk}^2) & (\xi_{jk}^1)(\xi_{jk}^2)^2 & (\xi_{jk}^1)^3(\xi_{jk}^2) & (\xi_{jk}^1)(\xi_{jk}^2)^3 & (\xi_{jk}^1)^2(\xi_{jk}^2)^2 \end{bmatrix} V_k \quad (74c)$$

$$\xi_{jk} = \left((\xi_{jk}^1)^2 + (\xi_{jk}^2)^2 \right)^{1/2} = \left((x_k^{nu} - x_j^{nu})^2 + (y_k^{nu} - y_j^{nu})^2 \right)^{1/2} \quad (74d)$$

$$V_k = \Delta x \times \Delta x \times 1 = \left(\frac{2}{n-1} \right)^2 \quad (74e)$$

where superscript nu represents the non-uniform particle. The point number in x or y direction is denoted by n in Eq.(74e), and Δx is the point spacing. Particle k represents the family member of particle j belonging to the non-uniform distributed system. N_j is the total number of the family members of particle j which belongs to the non-uniform distributed system. The terms ξ_{jk}^1 and ξ_{jk}^2 are defined as $\xi_{jk}^1 = x_k^{nu} - x_j^{nu}$ and $\xi_{jk}^2 = y_k^{nu} - y_j^{nu}$.

Subsequently, the function value on the position of particle i (shown in Fig. 16 (b)) can be predicted by using the derivatives on the position of particle j by using Taylor series expansion as (as shown in Eq. (B.4))

$$\begin{aligned}
\left(f(x_i^u, y_i^u)\right)^{corrected} &= f(x_j^{nu}, y_j^{nu}) + f_{,x}^{PDDO}(x_j^{nu}, y_j^{nu})(x_i^u - x_j^{nu}) \\
&+ f_{,y}^{PDDO}(x_j^{nu}, y_j^{nu})(y_i^u - y_j^{nu}) + \frac{1}{2} f_{,xx}^{PDDO}(x_j^{nu}, y_j^{nu})(x_i^u - x_j^{nu})^2 \\
&+ \frac{1}{2} f_{,yy}^{PDDO}(x_j^{nu}, y_j^{nu})(y_i^u - y_j^{nu})^2 + f_{,xy}^{PDDO}(x_j^{nu}, y_j^{nu})(x_i^u - x_j^{nu})(y_i^u - y_j^{nu})
\end{aligned} \tag{75}$$

where superscript u represents the uniform particle and nu represents the non-uniform particle. By using Eq. (75), and by using the non-uniform family members (j) located at (x_j^{nu}, y_j^{nu}) of particle (i) located at (x_i^u, y_i^u) , one corrected function value at (x_i^u, y_i^u) can be obtained for each family of (i) (such as (j)). Therefore, the predicted function value at (x_i^u, y_i^u) can be obtained by averaging the corrected values as

$$\begin{aligned}
\left(f(x_i^u, y_i^u)\right)^{predicted} &= \frac{\sum_{j=1}^{N_{i,nu}} \left(f(x_i^u, y_i^u)\right)^{corrected}}{N_{i,nu}} \\
&= \frac{\sum_{j=1}^{N_{i,nu}} \left(\begin{aligned} &f(x_j^{nu}, y_j^{nu}) + f_{,x}^{PDDO}(x_j^{nu}, y_j^{nu})(x_i^u - x_j^{nu}) \\ &+ f_{,y}^{PDDO}(x_j^{nu}, y_j^{nu})(y_i^u - y_j^{nu}) + \frac{1}{2} f_{,xx}^{PDDO}(x_j^{nu}, y_j^{nu})(x_i^u - x_j^{nu})^2 \\ &+ \frac{1}{2} f_{,yy}^{PDDO}(x_j^{nu}, y_j^{nu})(y_i^u - y_j^{nu})^2 + f_{,xy}^{PDDO}(x_j^{nu}, y_j^{nu})(x_i^u - x_j^{nu})(y_i^u - y_j^{nu}) \end{aligned} \right)}{N_{i,nu}}
\end{aligned} \tag{76}$$

where $N_{i,nu}$ is the total number of the non-uniform family members of particle located at (x_i^u, y_i^u) .

The function values assigned to the non-uniform particles, $f(x_j^{nu}, y_j^{nu})$ and predicted for the uniform particles, $\left(f(x_i^u, y_i^u)\right)^{predicted}$ are plotted in Fig. 17 and Fig. 18 for resolutions 21×21 and 51×51 , respectively.

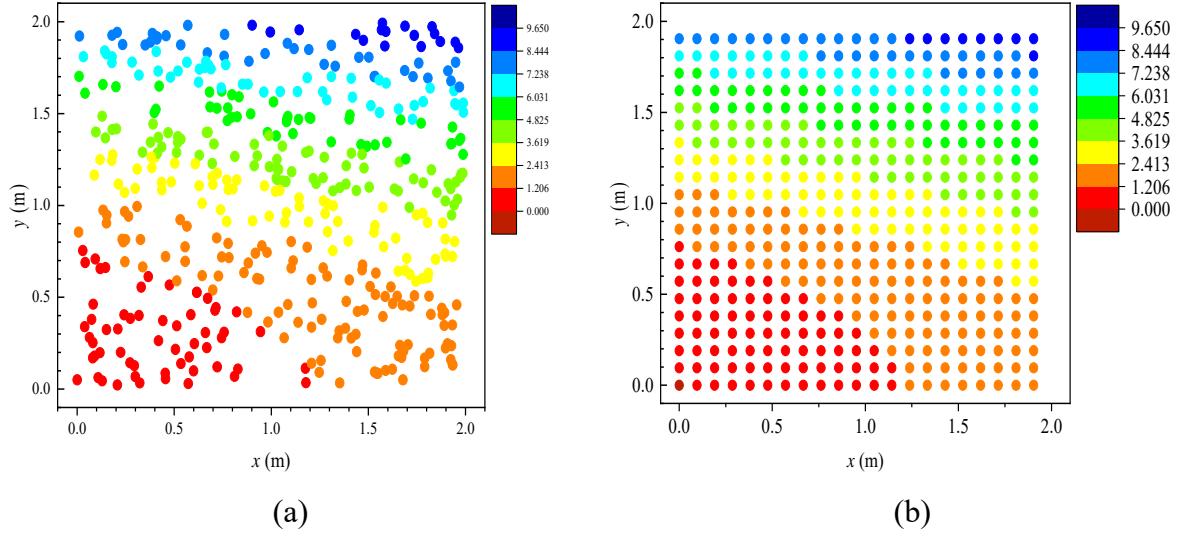


Fig. 17 Function value, for (a) non-uniform particles (assigned), $f(x_j^{nu}, y_j^{nu})$ and (b) uniform

particles (predicted), $(f(x_i^u, y_i^u))^{predicted}$ for resolution 21×21

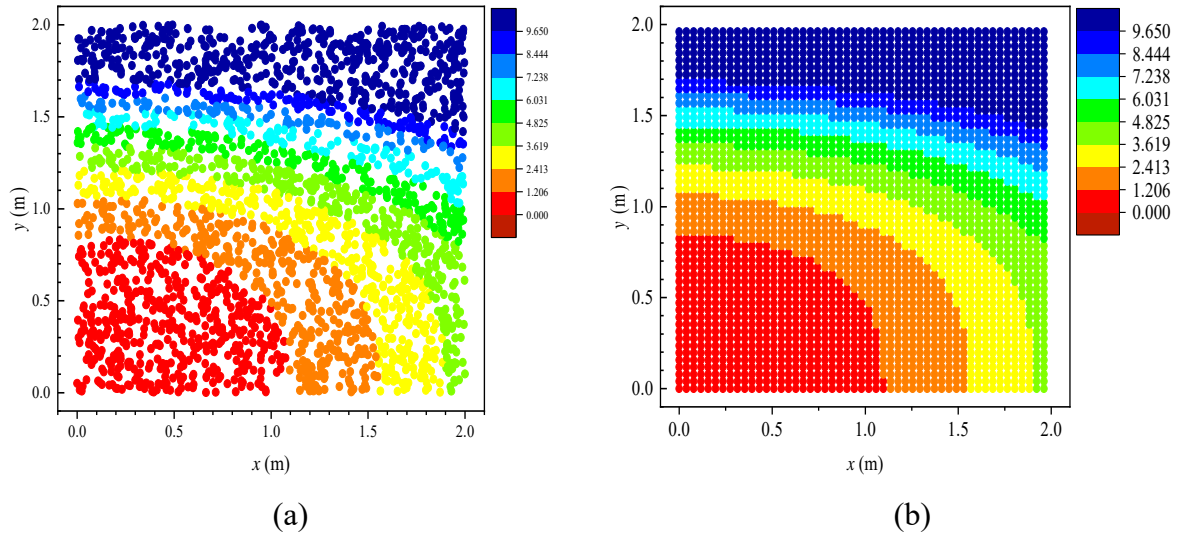


Fig. 18 Function value for (a) non-uniform particles (assigned), $f(x_j^{nu}, y_j^{nu})$ and (b) uniform

particles (predicted), $(f(x_i^u, y_i^u))^{predicted}$ for resolution 51×51

The total L_2 -norm error of the uniform particle set is calculated as

$$\varepsilon_r = \sqrt{\frac{\sum_i (f^{predicted}(x_i^u, y_i^u) - f^{exact}(x_i^u, y_i^u))^2}{\sum_i (f^{exact}(x_i^u, y_i^u))^2}} \quad (77)$$

By comparing the error for different resolutions as $\varepsilon_r = 0.31\%$ for 21×21 , $\varepsilon_r = 0.0670\%$ for 41×41 , and $\varepsilon_r = 0.0374\%$ for 51×51 , as shown in Fig. 19, it can be concluded that a higher resolution results in a more accurate solution.

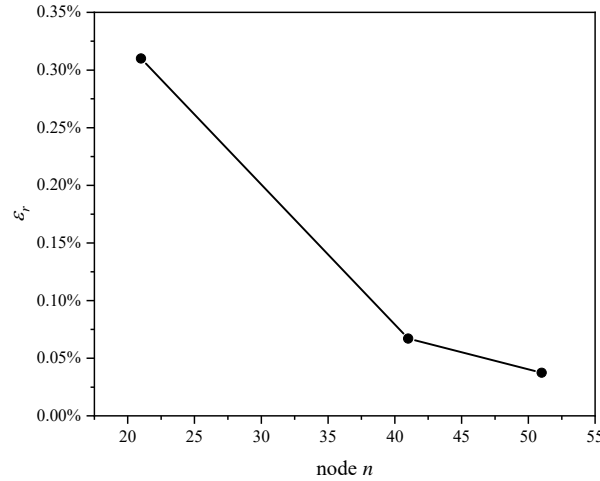


Fig. 19 L_2 -norm error plot for different resolutions

Example 5.1.1 represents the approximation of derivatives by using PDDO for uniform and non-uniform particle distributions. The derivatives are calculated by using function values at particle locations. On the other hand, example 5.1.2 represents the approximation of functions by using PDDO for uniform and non-uniform particle distributions. The functions are calculated by using approximated derivative values at particle locations. It can be concluded from the two tests that the accuracy of the PD differential operator both for approximating the function derivatives and function values is acceptable. Besides, although the irregularity of the particle distribution reduces the accuracy of the PDDO for predicting the derivatives, the PDDO can still be adopted to predict the function values based on these approximated differentials with acceptable accuracy.

5.2 Hydrostatic test

In the section, a hydrostatic simulation is conducted to study the energy conservation of the current PDDO model (Chen et al., 2015). The stationary fluids with their properties being $\rho_{0,A} = 3000 \text{ kg/m}^3$, $\rho_{0,B} = 1000 \text{ kg/m}^3$ and $\mu_A = \mu_B = 10^{-3} \text{ kg/ms}$ (Guo et al., 2018) are in an open container with dimensions being $L \times W = 0.05 \text{ m} \times 0.03 \text{ m}$, as shown in Fig. 20 (a). The gravitational acceleration is $|\mathbf{g}| = 9.8 \text{ m/s}^2$. For fluid particle i , it is implemented as

$$\mathbf{F}_i^B = \begin{bmatrix} 0 \\ -\rho_i |\mathbf{g}| \end{bmatrix} \quad (78)$$

The maximum velocity of the fluid is estimated as $|\mathbf{V}_{\max}| = \sqrt{2|\mathbf{g}|W} \approx 0.767$ m/s (Monaghan, 1994) and the material constants are $\gamma_A = \gamma_B = 1$. The artificial speeds of sound are hence chosen as 10 times of the maximum velocity as $c_A = c_B = 10|\mathbf{V}_{\max}| = 7.67$ m/s (Monaghan and Rafiee, 2013). The background pressure is set as $p_0 = 0$ since the free surface is involved (Krimi et al., 2018). As shown in Fig. 20 (b), 100×60 PDDO particles are distributed in a uniform form with the initial particle spacing is set as $\Delta x = 5 \times 10^{-4}$ m. The horizon size is chosen as $\delta = 3.6\Delta x$. In order to save the computational time, only one fictitious layer is imposed to simulate the solid wall and its density is assumed to be $\rho_{0,A}$. The solid wall boundary conditions are

$$x=0, x=L, y=0: v_x = v_y = 0 \quad (79)$$

where v_x and v_y are the horizontal and vertical components of velocity. Consequently, the boundary conditions are implemented by forcing the particles in the fictitious layer having the velocities calculated according to Eq.(53) with $\mathbf{v}_{wall} = 0$. In addition, the pressures of the fictitious solid particles are calculated according to Eq.(54a).

The real fluid particles on the free surface are forced to have the constraint as

$$y = W : P = 0 \quad (80)$$

The velocities of the free surface fluid particles are updated according to Eq.(47) and Eq.(50a).

The simulation time step size is chosen as $\Delta t = 5 \times 10^{-6}$ s with total simulation time as $t = 1.0$ s. The MLS algorithm is adapted for smoothing velocity and density fields for every 20 iterations, i.e. $n_{MLS} = 20$ (Colagrossi and Landrini, 2003). On the other hand, the XSPH modification in section 4.5 with the constant being $\chi = 0.08$ (Suresh et al., 2019a) is used for displacement fields smoothing. The Velocity Verlet algorithm in section 4.1 is adopted. The PST is not applied in this case. The surface tension force is not included since it is neglectable compared with the gravity force. The artificial viscosity is not used.

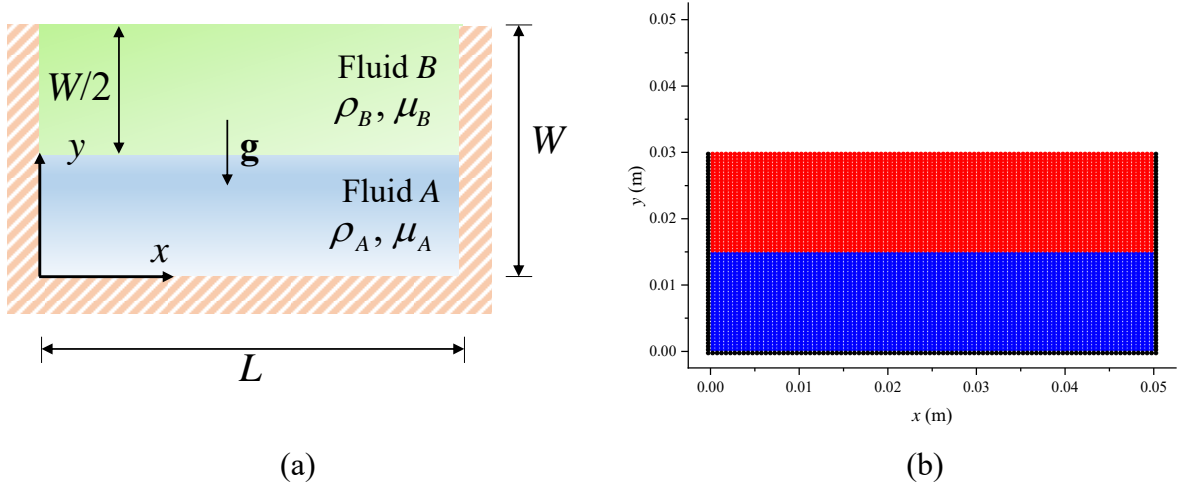


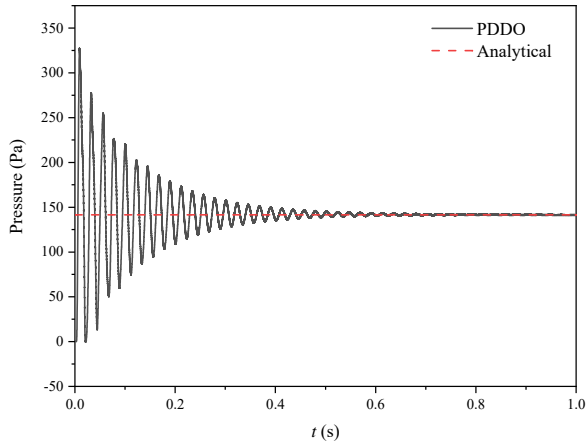
Fig. 20 Schematic of hydrostatic problem: (a) Geometry illustration and (b) PDDO discretization

The analytical solution for hydrostatic pressure is calculated as

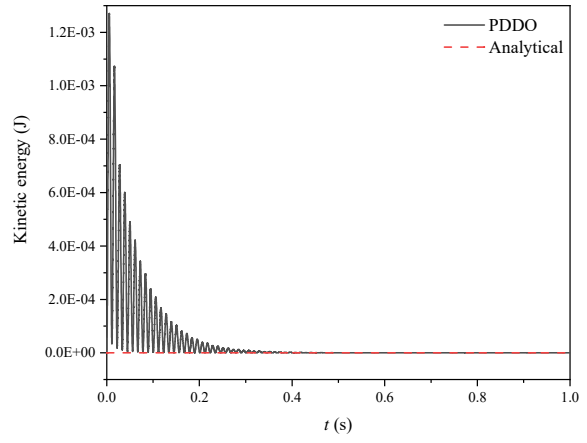
$$p_{B,analytical} = \rho_B |\mathbf{g}| (W - y) \quad (81)$$

$$p_{A,analytical} = \rho_B |\mathbf{g}| (W/2) + \rho_A |\mathbf{g}| (W/2 - y) \quad (82)$$

where y represents the vertical coordinate. The time histories of the predicted pressure and kinetic energy at the interface point $(x, y) = (0.02475, 0.01525)$ are presented in Fig. 21. The pressure and velocity magnitude distributions in the deformed configuration at the final time are presented in Fig. 22. It can be observed from Fig. 22 (b) that the velocities of the lighter fluid particles are not zero in the final stage. The largest velocity magnitude at the final simulation time is observed to be 1.76×10^{-4} (m/s) at the middle of the free surface, which is still in a relatively negligible level. Besides, a convergence study is conducted for PD nodes 40×24 , 80×48 , and 100×60 . The PDDO results are compared with analytical solutions. The pressure profile along the mid vertical line $x = 2.5 \times 10^{-2}$ m is compared with the analytical solution, as shown in Fig. 23. It can be observed that the PDDO results agree well with the analytical solutions. Thus, the capability of the present PDDO model for solving the two-phase hydrostatic problem is demonstrated. Besides, the energy conservation of the model is also validated.

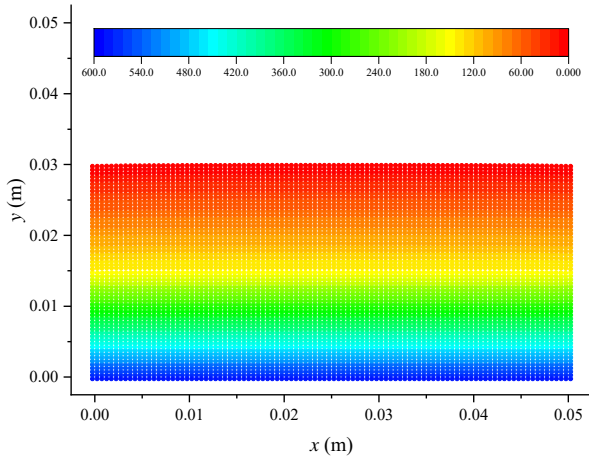


(a)

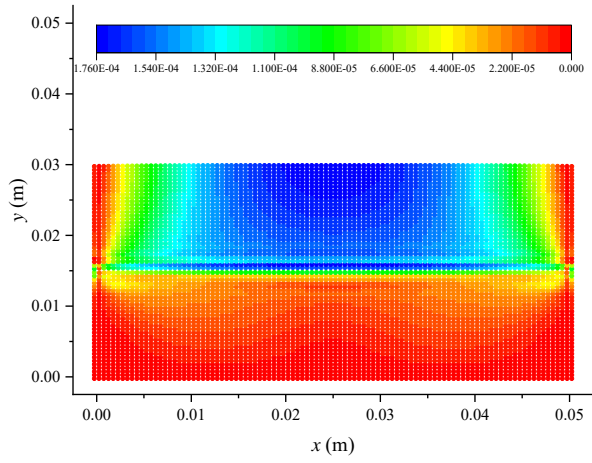


(b)

Fig. 21 Time history of (a) pressure and (b) kinetic energy at $(x, y) = (0.02475, 0.01525)$.



(a)



(b)

Fig. 22 (a) Pressure distribution and (b) velocity magnitude distribution on deformed configuration at $t = 1$ s

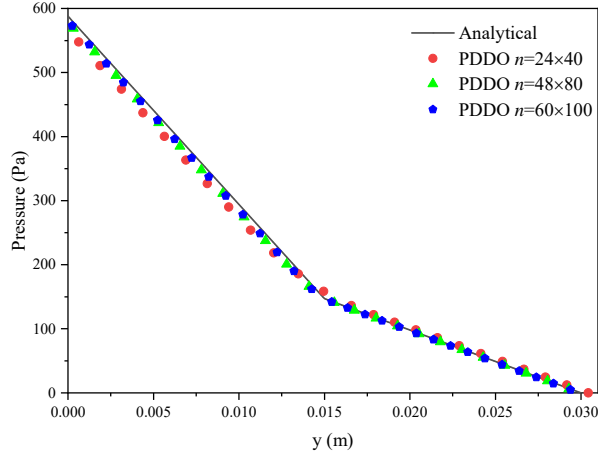


Fig. 23 Pressure variations of different resolutions on the vertical line $x = 2.5 \times 10^{-2}$ m at $t = 1$ s

5.3 Two-phase Poiseuille flow

The two-phase fluids laminar flow within two infinite plates has been studied in many work (Cao et al., 2004; Drouet et al., 2005; Gemmell and Epstein, 1962; Guo et al., 2018; Li and Renardy, 2000; Rezavand et al., 2018; Talon and Meiburg, 2011) as a benchmark problem in the field of multi-phase fluid flows. As shown in Fig. 24, two adjacent immiscible fluids Poiseuille laminar flow is simulated by using the developed PDDO model. The flow domain has the dimensions as $L = 2b = 10^{-3}$ m . The top and bottom boundaries are stationary and non-slip. The boundary conditions are

At $y = -b$

$$v_x = v_y = 0 \quad (83)$$

At $y = b$

$$v_x = v_y = 0 \quad (84)$$

Consequently, the velocities of the fictitious solid particles are calculated according to Eq.(53) with $\mathbf{v}_{wall} = 0$. In addition, the pressures of the fictitious solid particles are calculated according to Eq.(54a).

The initial conditions for all particles including the fluid particles and the fictitious solid particles are:

$$v_x = v_y = 0 \quad (85)$$

A pressure gradient is applied as a body force per unit mass as $F = 2 \times 10^{-4} \text{ m/s}^2$ over both fluids in the x direction, as for particle i

$$\mathbf{F}_i^B = \begin{bmatrix} \rho_i F \\ 0 \end{bmatrix} \quad (86)$$

The densities of both fluids are the same as $\rho_{0,A} = \rho_{0,B} = 1000 \text{ kg/m}^3$. The plates are assumed to have the same density. The viscosity coefficient of fluid B is fixed as $\mu_B = 10^{-3} \text{ kg/ms}$. The viscosity coefficient of fluid A has three different values, namely, $2 \times 10^{-3} \text{ kg/ms}$, $5 \times 10^{-3} \text{ kg/ms}$ and $10 \times 10^{-3} \text{ kg/ms}$. The material constants both for fluid A and B are set as $\gamma_A = \gamma_B = 1$. The maximum velocity of the fluid is set as $|\mathbf{V}_{\max}| = 8.7 \times 10^{-6} \text{ m/s}$ (Rezavand et al., 2018). Hence, the artificial speeds of sound become $c_A = c_B = 8.7 \times 10^{-5} \text{ m/s}$ (Monaghan, 1992). The background pressure is not adopted for this simple laminar fluid flow case. In the PDDO simulation, an initial particle spacing of $\Delta x = 1 \times 10^{-5} \text{ m}$ is adopted. The time step size is chosen as $\Delta t = 1 \times 10^{-5} \text{ s}$. The total simulation time is chosen as $t = 0.6 \text{ s}$ to achieve a steady state at the end of the simulation. The fluids are stationary at the initial state. Since the surface tension effect is small and neglectable compared to the applied body force, no surface tension force is applied. The PST, the MLS, the XSPH, and the artificial viscosity are not used in this case. The Euler forward scheme is used.

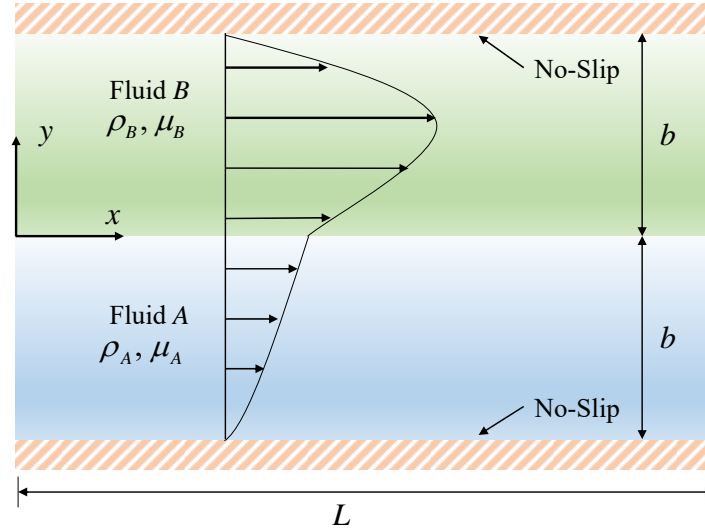


Fig. 24 Schematic of two-phase Poiseuille flow

The analytical solutions of the horizontal velocity for the two fluids are provided by Bird et al. (Bird et al., 2007) as

$$v_{A,x} = \frac{\rho_A F b^2}{2\mu_A} \left[\left(\frac{2\mu_A}{\mu_A + \mu_B} \right) + \left(\frac{\mu_A - \mu_B}{\mu_A + \mu_B} \right) \left(\frac{y}{b} \right) - \left(\frac{y}{b} \right)^2 \right] \quad (87a)$$

$$v_{B,x} = \frac{\rho_B F b^2}{2\mu_B} \left[\left(\frac{2\mu_B}{\mu_A + \mu_B} \right) + \left(\frac{\mu_A - \mu_B}{\mu_A + \mu_B} \right) \left(\frac{y}{b} \right) - \left(\frac{y}{b} \right)^2 \right] \quad (87b)$$

where $v_{A,x}$ and $v_{B,x}$ represent the horizontal velocity for fluid A and B respectively. The term y represents the vertical coordinate. The analytical average velocities are calculated as (Bird et al., 2007)

$$\bar{v}_{A,x} = \frac{1}{b} \int_{-b}^0 v_{A,x} dy = \frac{\rho F b^2}{12\mu_A} \left(\frac{7\mu_A + \mu_B}{\mu_A + \mu_B} \right) \quad (88a)$$

$$\bar{v}_{B,x} = \frac{1}{b} \int_0^b v_{B,x} dy = \frac{\rho F b^2}{12\mu_B} \left(\frac{\mu_A + 7\mu_B}{\mu_A + \mu_B} \right) \quad (88b)$$

$$\bar{v}_x = (\bar{v}_{A,x} + \bar{v}_{B,x})/2 \quad (88c)$$

where $\bar{v}_{A,x}$ and $\bar{v}_{B,x}$ represent the average horizontal velocity for fluids A and B respectively. The average horizontal velocity over the whole flow domain, i.e. \bar{v}_x , is calculated according to Eq.(88c).

The comparison between the PDDO and analytical solutions of the normal velocity v_x / \bar{v}_x at the steady-state is provided in Fig. 25. It can be inferred from the figure that the results agree well.

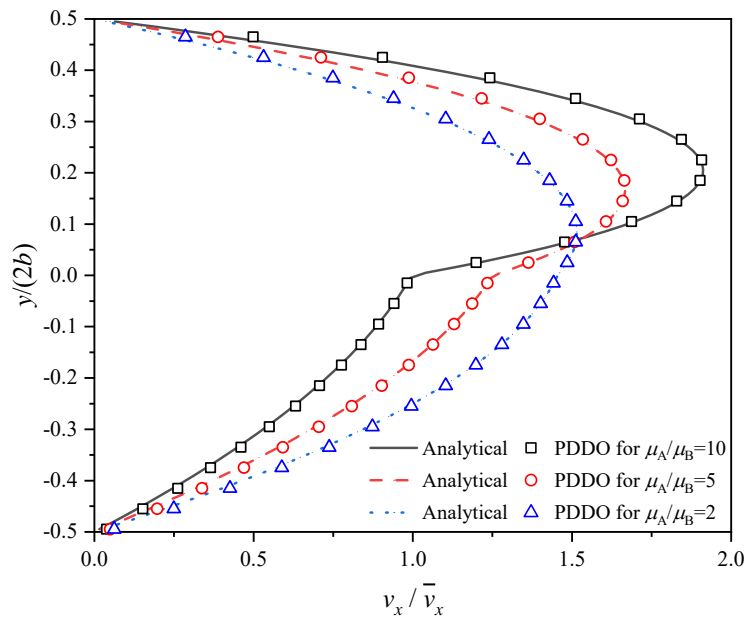


Fig. 25 Comparison of the horizontal velocity profile between the PDDO results and analytical solutions in three different viscosity coefficient ratios at $t = 0.6$ s .

The velocity distributions in three different viscosity coefficient ratios at time $t = 0.6$ s are provided in Fig. 26. Besides, the pressures are zero in the fluid field in these three cases because the gravity effect is not considered, and the driving forces are applied as body forces.

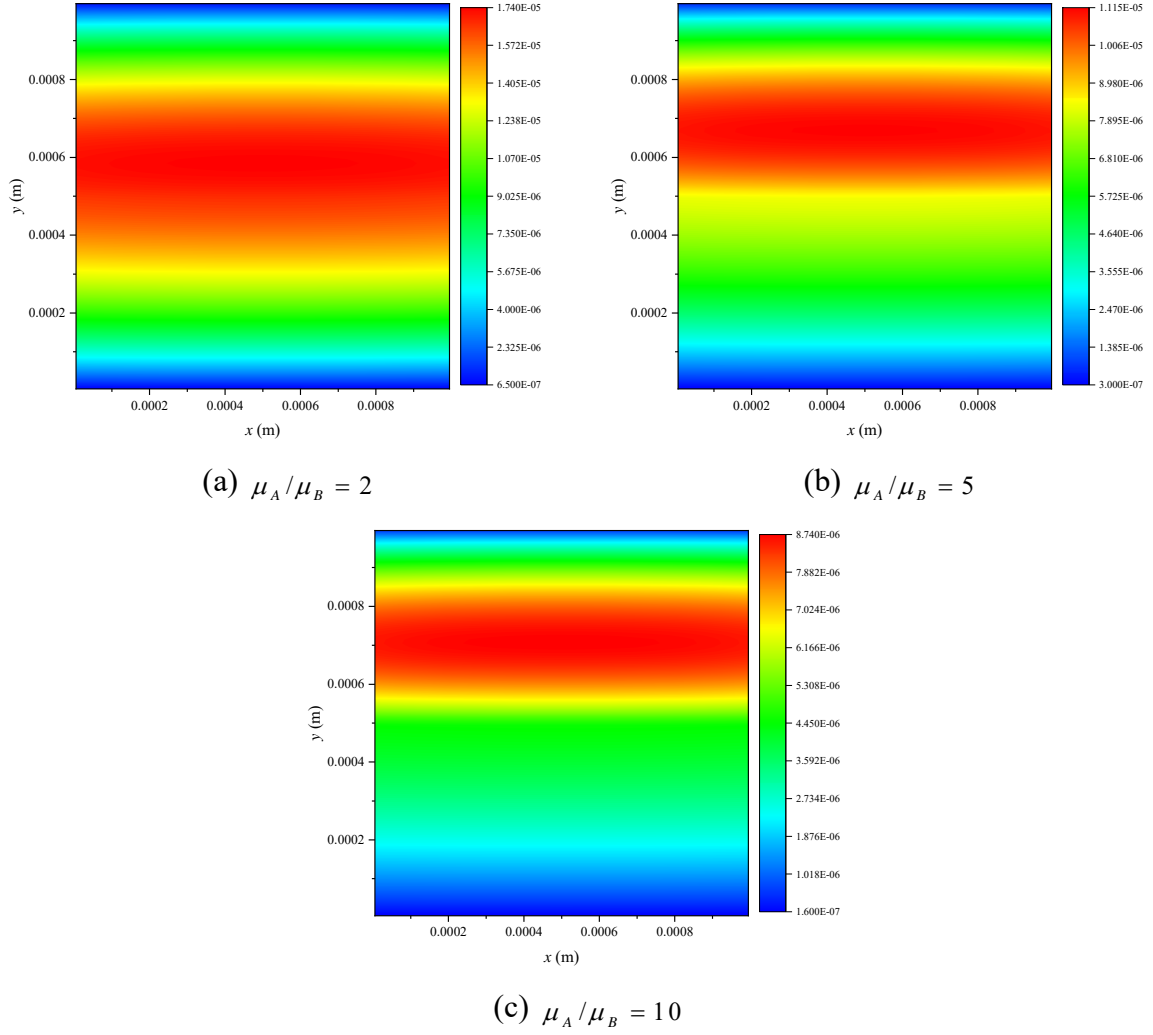


Fig. 26 The contour plots of the horizontal velocity distributions in three different viscosity coefficient ratios at $t = 0.6$ s .

A convergence study is also carried out for different resolutions and viscosity coefficients, namely $\Delta x = 5 \times 10^{-5}$ m, 3.3×10^{-5} m, 2×10^{-5} m, 1×10^{-5} m for $\mu_A/\mu_B = 10, 5, 2$, respectively.

The error can be calculated as

$$\mathcal{E}_r = \sqrt{\frac{\sum_i (v_{x,PDDO} - v_{x,analytical})^2}{\sum_i (v_{x,analytical})^2}} \quad (89)$$

From Fig. 27, it can be observed that the numerical simulation error decreases with increasing PDDO particles. Furthermore, the error becomes larger with larger viscosity ratios. The peridynamic differential operator used in this study has a third order accuracy, which is based on second order Taylor series expansion. But in this case, the Euler method used for velocity

prediction only has first order accuracy, leading to a reduction of accuracy for velocity prediction.

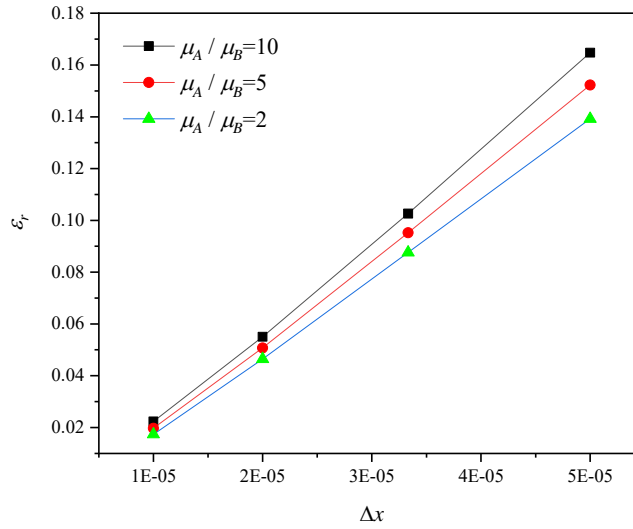


Fig. 27 Rate of convergence for two-phase Poiseuille flow in three viscosity ratios

5.4 2D droplet deformation

To validate the surface tension model, a 2D square droplet deformation simulation is conducted as previous ones (Adami et al., 2010; Guo et al., 2018; Krimi et al., 2018; Yan et al., 2019). A square droplet containing fluid A with sides $l_d = 0.6$ m is located in the centre of a container with its sides being $L = 1$ m, as shown in Fig. 28 (a). The droplet is surrounded by fluid B. The two fluids have the same density and viscosity coefficient as $\rho_A = \rho_B = 1 \text{ kg/m}^3$ and $\mu_A = \mu_B = 0.2 \text{ Pa} \cdot \text{s}$. The density of the solid wall is assumed to be ρ_B . The surface tension coefficient is set as $\beta = 1 \text{ N/m}$ (Adami et al., 2010; Guo et al., 2018; Krimi et al., 2018; Yan et al., 2019).

The initial condition is

$$\text{at } t = 0 : \mathbf{a} = 0, \mathbf{v} = 0, \mathbf{u} = 0 \quad (90)$$

The non-slip solid boundary condition is applied. As shown in Fig. 28 (b), the fluid domain is discretized by 80×80 particles and three layers of the fictitious particles are imposed to represent the fixed solid boundary as

$$x = -L/2, x = L/2, y = -W/2, y = W/2: v_x = v_y = 0 \quad (91)$$

Therefore, the velocities of the fictitious solid particles are calculated according to Eq.(53) with $\mathbf{v}_{wall} = 0$. The pressure of the fictitious solid particles is calculated according to Eq.(54a). On the fluids interface, the surface tension force is included.

The horizon size is chosen as $\delta = 3.6\Delta x$. The speed of the sound is set as $c_A = c_B = 10$ m/s and the material constants are $\gamma_A = \gamma_B = 1$. The maximum velocity magnitude thus can be calculated as $|\mathbf{V}_{max}| = c_A/10 = 1$ m/s (Monaghan, 1992). Furthermore, the background pressure is calculated as $P_0 = 0.05P_{ref} = 0.05 \frac{\rho_{0,B}c_A^2}{\gamma_A} = 5$ Pa as explained in Section 3.1. The time step size is chosen as $\Delta t = 1 \times 10^{-4}$ s and the total simulation time is $t = 1$ s.

The MLS algorithm is performed every 20 iterations, i.e. $n_{MLS} = 20$ (Colagrossi and Landrini, 2003) for smoothing velocity and density fields. The PST without pressure and velocity correction is performed as well for smoothing displacement fields. The constant in PST is chosen as $C_{PST} = 0.01$. Subsequently, it can be calculated according to Eq.(60) as $\alpha_{PST} = 1 \times 10^{-4}$ m. The artificial viscosity in Section 4.6 is adopted to make the simulation be more stable. The velocity Verlet algorithm is used. The XSPH is not used in this case.

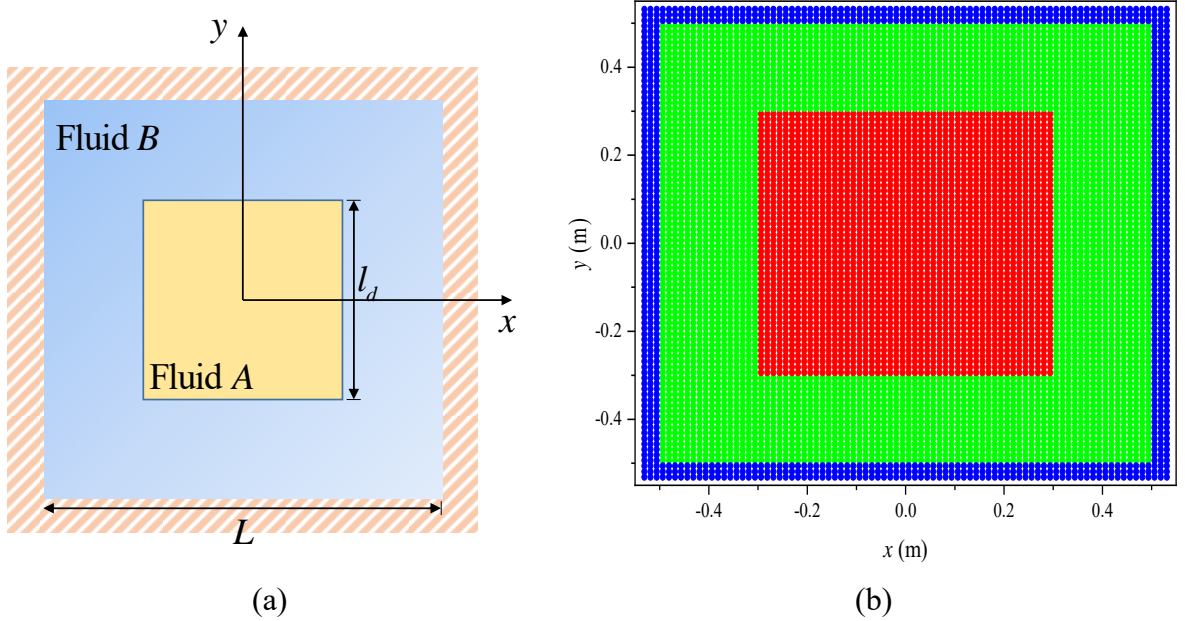


Fig. 28 2D square droplet deformation under surface tension force (a) geometry sketch and (b) PDDO discretization with 80×80 nodes

The evolution of the droplet configuration is provided in Fig. 29. It can be observed from the figure that at the equilibrium state, the square droplet is transformed in a circular shape. The radius of the final circular droplet can be calculated as $R = l_d / \sqrt{\pi} \approx 0.338 \text{ m}$ ($\pi R^2 = l_d^2$ with incompressible hypothesis (Adami et al., 2010; Guo et al., 2018; Krimi et al., 2018; Yan et al., 2019)). Consequently, the pressure drop between the two phases is calculated according to Laplace's law as (Chen et al., 2015)

$$\Delta p = \beta \kappa = \frac{\beta}{R} \approx 2.954 \text{ Pa} \quad (92)$$

where the inner fluid has a higher pressure to balance the surface tension force. The pressure profile along the horizontal line $y = 0$ with different resolutions is provided in Fig. 30(a). The time history of the average pressure difference $\Delta P = \bar{P}_A - \bar{P}_B$ is provided in Fig. 30 (b) where \bar{P}_A and \bar{P}_B represent the average pressure of the fluid particles for fluid A and B which are not in the interface region, respectively. The pressure distribution is provided in Fig. 30(c). Due to the viscosity force, the kinetic energy of the square droplet will decrease with time increasing, leading to a steady state finally. The time history of the kinetic energy and the velocity distribution at the final stage are provided in Fig. 31. With the refinement of the resolution, the magnitude of the kinetic energy decreases at the quasi-equilibrium state, within the 1×10^{-5} level for 80×80 particles. However, it can also be observed from Fig. 31 (b) that even for the quasi-equilibrium state, the oscillation of the interface still exists. This oscillation, also can be noted as the spurious current, may be created by the slight variation of curvature because of the discrete nature of numerical approach according to the explanation given in (Remington et al., 1995). The L_1 norm (Remington et al., 1995) defined as $L_1 = \sum_1^{N_{node}} |\mathbf{v}| / N_{node}$ is 3.3×10^{-3} , and L_2 norm (Remington et al., 1995) defined as $L_2 = \sqrt{\sum_1^{N_{node}} |\mathbf{v}|^2 / N_{node}^2}$ is 7.9×10^{-5} at the final time. Besides, it can be observed in Fig. 29 that the interface near the square corners is not smooth, which may be the result of the utilization of PST. The shifting distance of PST is calculated only based on the position of the points (Eqs.(59)-(61)), regardless of the fluid phases. Thus, it may lead to the less smoothness at the interface (Mokos et al., 2017).

Modifications of the PST for multi-phase fluid flow in the PDDO form should be done as a future work.

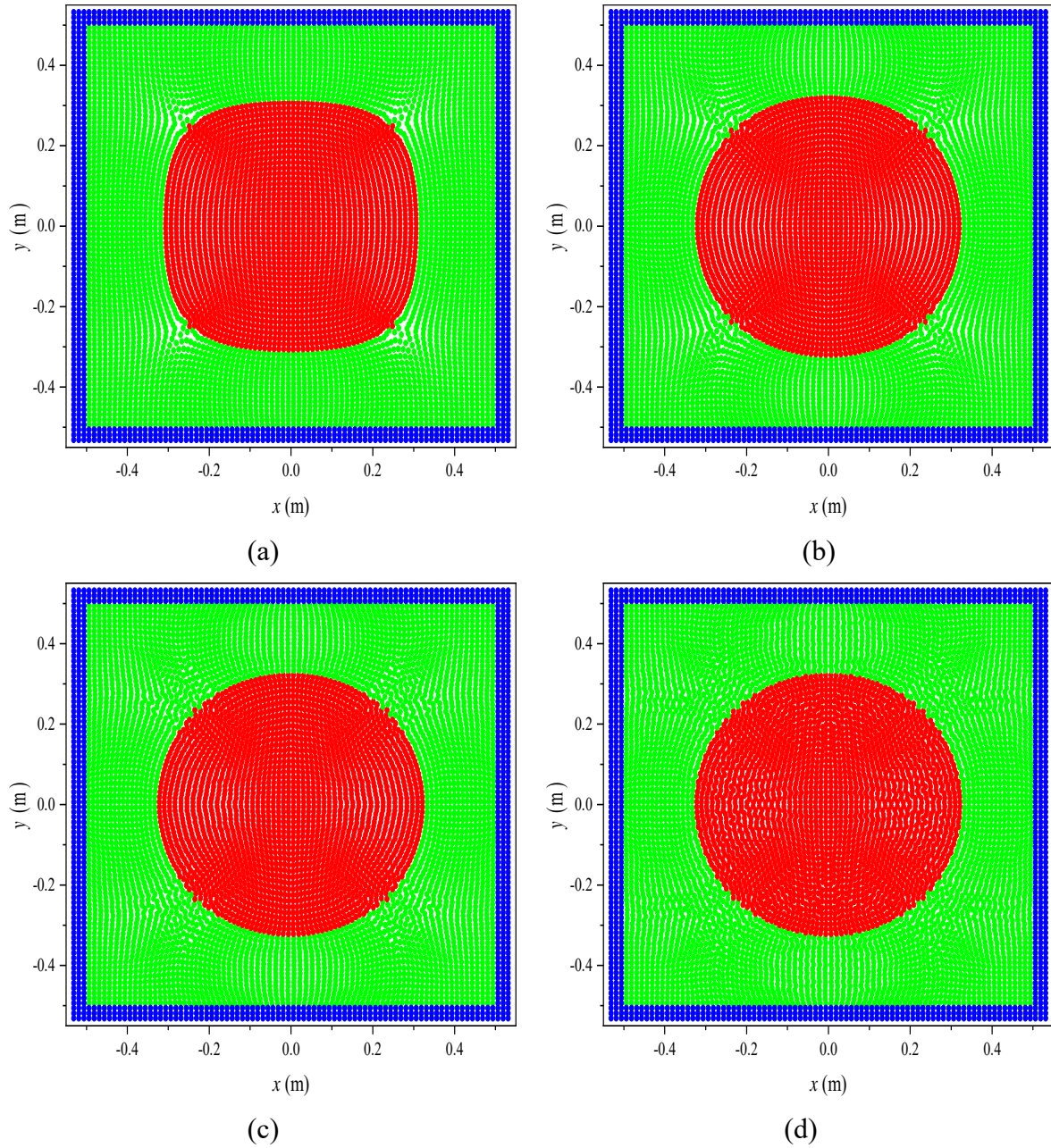


Fig. 29 Particle distribution at time (a) $t = 0.1$ s, (b) $t = 0.3$ s, (c) $t = 0.5$ s, and (d) $t = 1$ s with 80×80 particles

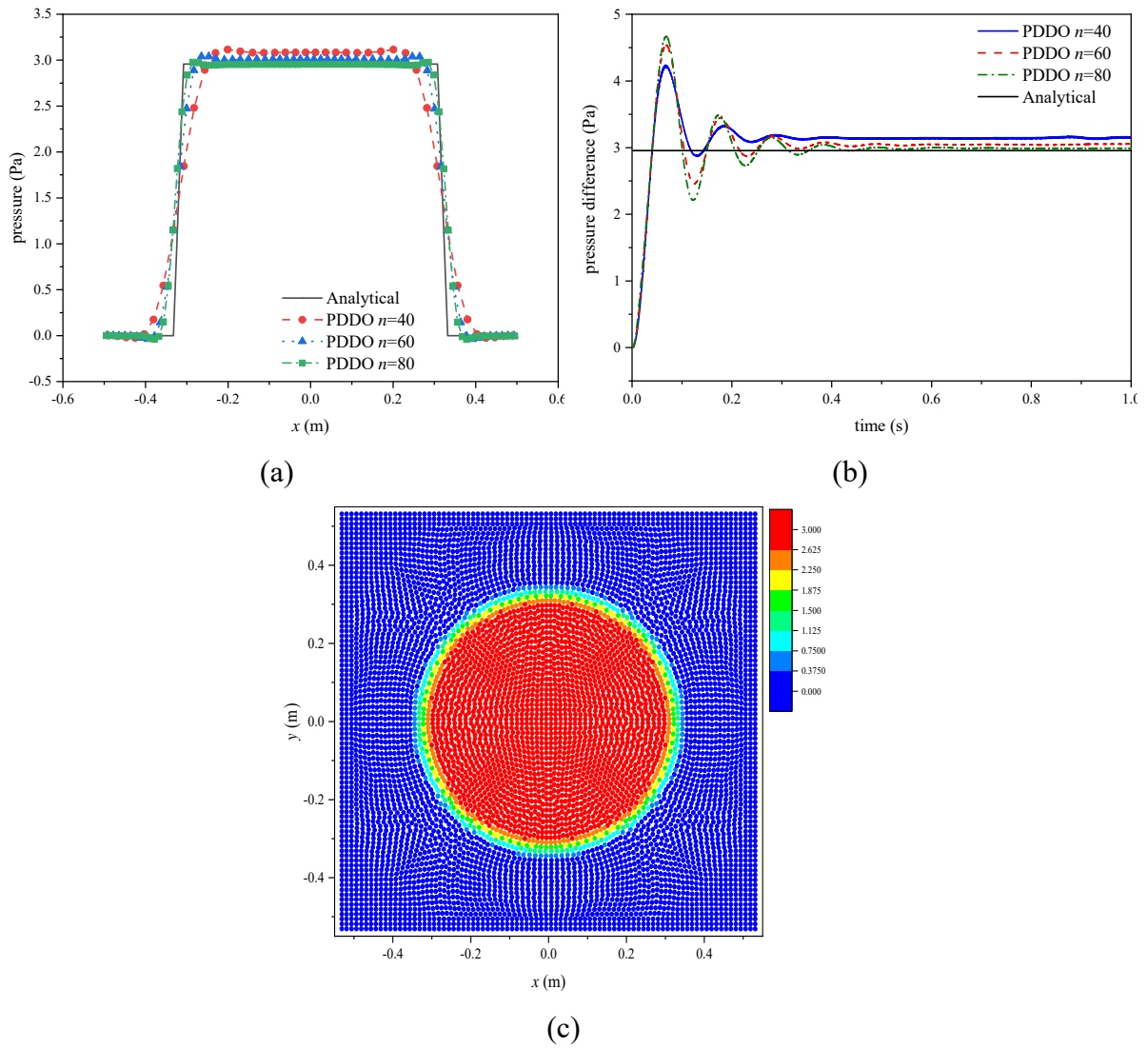


Fig. 30 Pressure profile (a) comparison with analytical solution along $y = 0$, (b) time history of the average pressure difference, and (c) pressure distribution with 80×80 particles

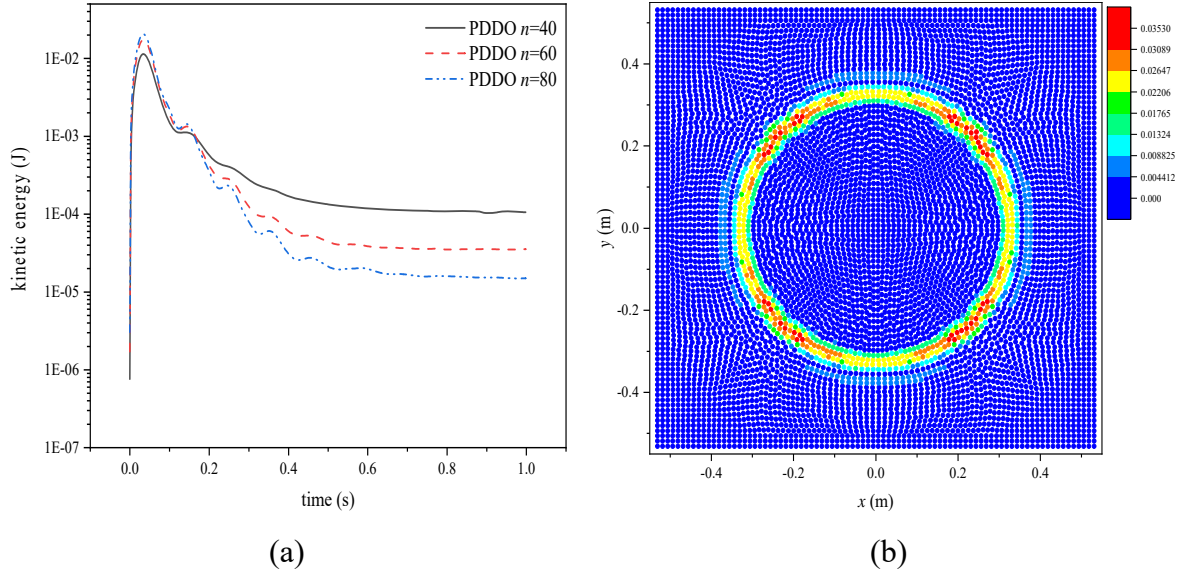


Fig. 31 (a) Time history of the kinetic energy of the inner droplet and (b) velocity distribution at the final state for 80×80 resolution

The orientation and magnitude of the surface tension force obtained by the current PDDO model are provided in Fig. 32 (a). Besides, the SPH result (Krimi et al., 2018) is provided in Fig. 32 (b) for comparison. It can be observed that all the surface tension forces are pointing to the centre of the inner droplet, being perpendicular to the interface. Because the interparticle index numbers used in this paper are always positive, the normal vector always points out to the other fluid phase. In other words, the normal vector of fluid particle A points to fluid B and the normal vector of fluid particle B points to fluid A. However, the curvatures on the two sides of the interface belonging to the two different fluid phases have opposite sign values, one being positive and the other being negative. As a result, the final surface tension forces always point to the centre. Also, the magnitude of the surface tension force decreases from the interface to the edge of the interface region.

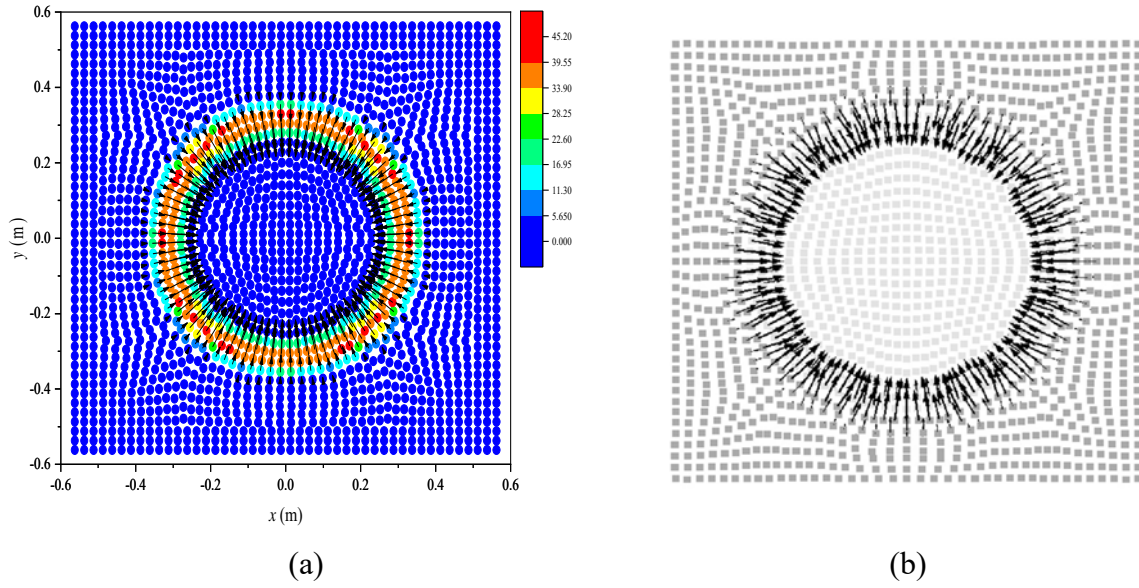


Fig. 32 The direction of the surface tension force and magnitude (a) obtained by PDDO for 40×40 nodes and (b) obtained by SPH 40×40 nodes (Krimi et al., 2018)

Furthermore, two more simulation cases are conducted: case 1 for $R_{ratio} = \mu_B / \mu_A = \rho_B / \rho_A = 10$, and case 2 for $R_{ratio} = \mu_B / \mu_A = \rho_B / \rho_A = 5$, as shown in Table 2.

Table 2 Density and viscosity values for different cases

Case number	ρ_A (kg/m ³)	ρ_B (kg/m ³)	μ_A (Pa·s)	μ_B (Pa·s)	R_{ratio}
0	1	1	0.2	0.2	1
1	1	5	0.2	1	5
2	1	10	0.2	2	10

The final configurations of the droplet for both cases are provided in Fig. 33 (a) and (b). The pressure profiles along $y = 0$ for $R_{ratio} = 1$, $R_{ratio} = 5$, and $R_{ratio} = 10$ are provided in Fig. 33 (c) for comparison. It is interesting to see that the case of $R_{ratio} = 5$ has the worst accuracy among all these three cases. According to the investigation in (Krimi et al., 2018), both a smaller viscosity coefficient ratio and a larger density ratio can give less accurate results. In the current study, the accuracy of case 2 which has a relatively smaller viscosity coefficient ratio and a smaller density ratio is less than the one of case 1 which has a relatively larger viscosity coefficient ratio and a larger density ratio. Therefore, in the present work, the ratio of

the viscosity coefficient has a larger effect than the ratio of density on the accuracy of the simulation results.

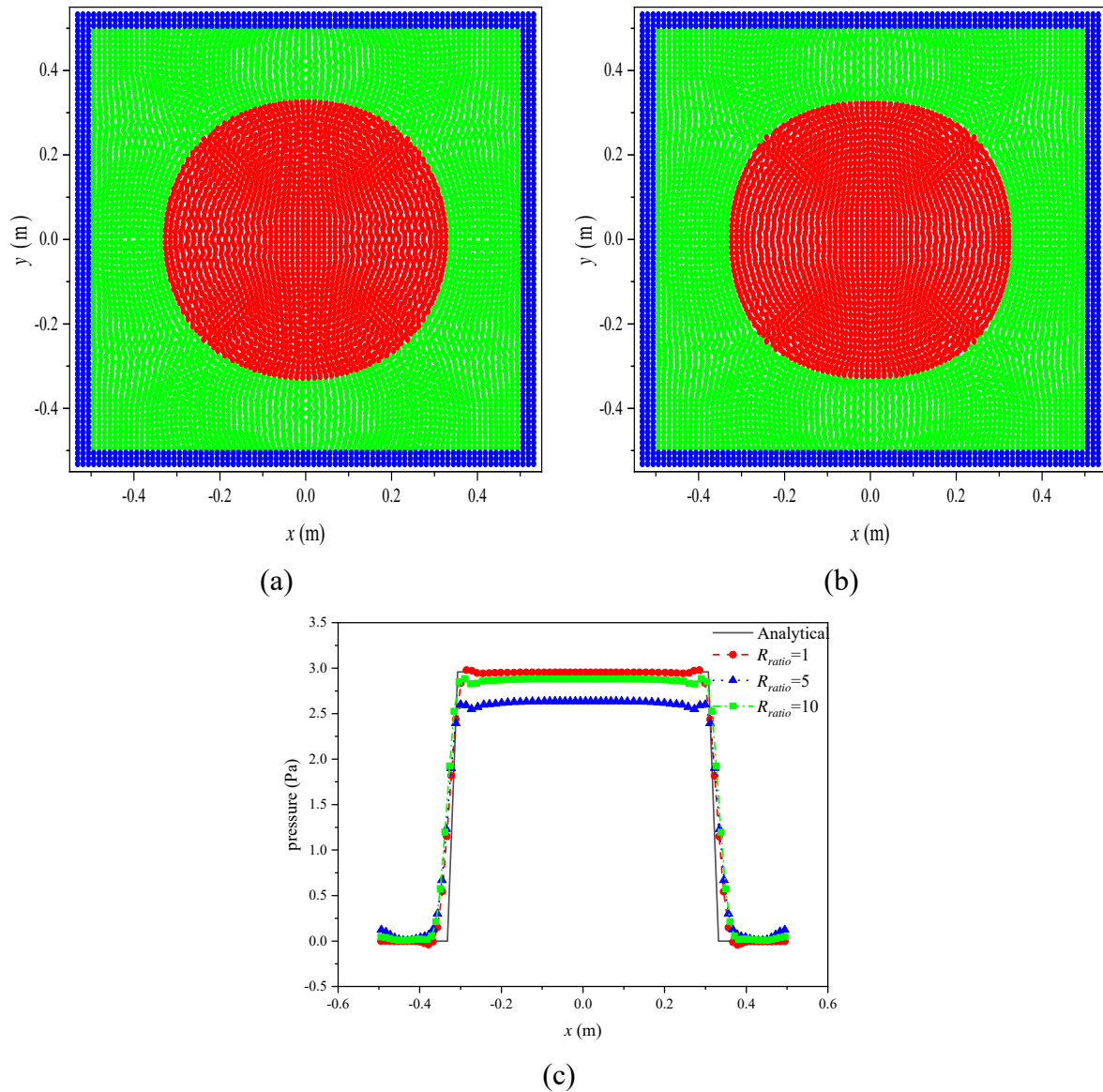


Fig. 33 Particle distributions at time $t = 1$ s with 80×80 particles for (a) case 1, and (b) case 2. (c) pressure profiles comparison

6 Conclusion

In this study, a multi-phase fluid laminar flow model is developed by using the peridynamic differential operator. The surface tension force originally from the Continuum Surface Force model is converted into a PDDO expression. The developed model is applied to solve the multi-phase fluid flow problems, e.g. two-phase hydrostatic problem, two-phase Poiseuille flow, 2D droplet deformation. The good agreements between the PDDO results and

the existing results demonstrate the capability of the present model for the multi-phase fluid flow simulations.

Acknowledgement

The authors gratefully acknowledge financial support from the China Scholarship Council (CSC No. 201506230126) and University of Strathclyde.

References

- Adami, S., Hu, X.Y., Adams, N.A., 2010. A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation. *Journal of Computational Physics* 229 (13), 5011-5021.
- Adami, S., Hu, X.Y., Adams, N.A., 2012. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics* 231 (21), 7057-7075.
- Adams, J.C., Brainerd, W.S., Martin, J.T., Smith, B.T., Wagener, J.L., 1997. *Fortran 95 handbook: complete ISO/ANSI reference*. MIT press.
- Akbari, H., 2019. An improved particle shifting technique for incompressible smoothed particle hydrodynamics methods. *International Journal for Numerical Methods in Fluids* 90 (12), 603-631.
- Bazazzadeh, S., Shojaei, A., Zaccariotto, M., Galvanetto, U., 2018. Application of the peridynamic differential operator to the solution of sloshing problems in tanks. *Engineering Computations* 36 (1), 45-83.
- Bird, R.B., Stewart, W.E., Lightfoot, E.N., 2007. *Transport phenomena*. John Wiley & Sons.
- Bobaru, F., Duangpanya, M., 2010. The peridynamic formulation for transient heat conduction. *International Journal of Heat and Mass Transfer* 53 (19-20), 4047-4059.
- Brackbill, J.U., Kothe, D.B., Zemach, C., 1992. A continuum method for modeling surface tension. *Journal of Computational Physics* 100 (2), 335-354.
- Cao, Q., Sarkar, K., Prasad, A.K., 2004. Direct numerical simulations of two-layer viscosity-stratified flow. *International Journal of Multiphase Flow* 30 (12), 1485-1508.
- Chen, Z., Zong, Z., Li, H.T., Li, J., 2013. An investigation into the pressure on solid walls in 2D sloshing using SPH method. *Ocean Engineering* 59, 129-141.
- Chen, Z., Zong, Z., Liu, M.B., Zou, L., Li, H.T., Shu, C., 2015. An SPH model for multiphase flows with complex interfaces and large density differences. *Journal of Computational Physics* 283, 169-188.
- Colagrossi, A., Bouscasse, B., Antuono, M., Marrone, S., 2012. Particle packing algorithm for SPH schemes. *Computer Physics Communications* 183 (8), 1641-1653.
- Colagrossi, A., Landrini, M., 2003. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics* 191 (2), 448-475.
- Cummins, S.J., Rudman, M., 1999. An SPH Projection Method. *Journal of Computational Physics* 152 (2), 584-607.
- Dilts, G.A., 1999. Moving-least-squares-particle hydrodynamics—I. Consistency and stability. *International journal for numerical methods in engineering* 44 (8), 1115-1155.
- Dilts, G.A., 2000. Moving least-squares particle hydrodynamics II: conservation and boundaries. *International journal for numerical methods in engineering* 48 (10), 1503-1524.
- Diyaroglu, C., Madenci, E., Oterkus, S., Oterkus, E., 2017. Moisture diffusion modelling by using peridynamics, ASME 2017 International Mechanical Engineering Congress & Exposition.

Diyaroglu, C., Madenci, E., Oterkus, S., Oterkus, E., 2018. A novel moisture diffusion modeling approach using finite element analysis. *Electronics* 7 (12), 438.

Dorduncu, M., 2019. Stress analysis of laminated composite beams using refined zigzag theory and peridynamic differential operator. *Composite Structures* 218, 193-203.

Drouet, L., Beltran, C., Edwards, N., Haurie, A., Vial, J.-P., Zachary, D.S.J.C.c., economic dynamics. Kluwer, t.a.i., 2005. An oracle method to couple climate and economic dynamics.

Fürstenau, J.-P., Avci, B., Wriggers, P., 2016. A Numerical Review of Multi-Fluid SPH Algorithms for High Density Ratios, *Advances in Computational Fluid-Structure Interaction and Flow Simulation*. Springer, pp. 139-150.

Gao, Y., Oterkus, S., 2018. Ordinary state-based peridynamic modelling for fully coupled thermoelastic problems. *Continuum Mechanics and Thermodynamics* 31 (4), 907-937.

Gao, Y., Oterkus, S., 2019a. Fully coupled thermomechanical analysis of laminated composites by using ordinary state based peridynamic theory. *Composite Structures* 207, 397-424.

Gao, Y., Oterkus, S., 2019b. Non-local modeling for fluid flow coupled with heat transfer by using peridynamic differential operator. *Engineering Analysis with Boundary Elements* 105, 104-121.

Gao, Y., Oterkus, S., 2019c. Nonlocal numerical simulation of low Reynolds number laminar fluid motion by using peridynamic differential operator. *Ocean Engineering* 179, 135-158.

Gemmell, A.R., Epstein, N., 1962. Numerical analysis of stratified laminar flow of two immiscible Newtonian liquids in a circular pipe. *The Canadian Journal of Chemical Engineering* 40 (5), 215-224.

Guo, K., Chen, R., Qiu, S., Tian, W., Su, G., 2018. An improved Multiphase Moving Particle Semi-implicit method in bubble rising simulations with large density ratios. *Nuclear Engineering and Design* 340, 370-387.

Han, S., Lim, S., Bae, J., Hwang, Y., Lee, S., Oterkus, S., Madenci, E., Diyaroglu, C., Oterkus, E., 2015. Equivalent acceleration assessment of JEDEC moisture sensitivity levels using peridynamics, 2015 IEEE 65th Electronic Components and Technology Conference (ECTC), pp. 1518-1523.

Heck, A., 1993. *Introduction to MAPLE*. Springer-Verlag New York.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 39 (1), 201-225.

Koshizuka, S., Oka, Y., 2017. Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid. *Nuclear Science and Engineering* 123 (3), 421-434.

Krimi, A., Rezoug, M., Khelladi, S., Nogueira, X., Deligant, M., Ramírez, L., 2018. Smoothed Particle Hydrodynamics: A consistent model for interfacial multiphase fluid flow simulations. *Journal of Computational Physics* 358, 53-87.

Li, J., Renardy, Y., 2000. Numerical Study of Flows of Two Immiscible Liquids at Low Reynolds Number. *Siam Review* 42 (3), 417-439.

Lind, S.J., Xu, R., Stansby, P.K., Rogers, B.D., 2012. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics* 231 (4), 1499-1523.

Liu, R., Yan, J., Li, S., 2019. Modeling and simulation of ice–water interactions by coupling peridynamics with updated Lagrangian particle hydrodynamics. *Computational Particle Mechanics* 7 (2), 241-255.

Liu, R.W., Xue, Y.Z., Lu, X.K., Cheng, W.X., 2018. Simulation of ship navigation in ice rubble based on peridynamics. *Ocean Engineering* 148, 286-298.

Madenci, E., Barut, A., Dorduncu, M., 2019. *Peridynamic differential operator for numerical analysis*. Springer.

Madenci, E., Barut, A., Futch, M., 2016. Peridynamic differential operator and its applications. *Computer Methods in Applied Mechanics and Engineering* 304, 408-451.

Madenci, E., Oterkus, E., 2014. *Peridynamic theory and its applications*. Springer.

Mokos, A., Rogers, B.D., Stansby, P.K., 2017. A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles. *Journal of Hydraulic Research* 55 (2), 143-162.

Monaghan, J.J., 1992. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30 (1), 543-574.

Monaghan, J.J., 1994. Simulating Free Surface Flows with SPH. *Journal of Computational Physics* 110 (2), 399-406.

Monaghan, J.J., Gingold, R.A., 1983. Shock simulation by the particle method SPH. *Journal of Computational Physics* 52 (2), 374-389.

Monaghan, J.J., Rafiee, A., 2013. A simple SPH algorithm for multi-fluid flow with high density ratios. *International Journal for Numerical Methods in Fluids* 71 (5), 537-561.

Morris, J.P., 2000. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids* 33 (3), 333-353.

Nangia, N., Griffith, B.E., Patankar, N.A., Bhalla, A.P.S., 2019a. A robust incompressible Navier-Stokes solver for high density ratio multiphase flows. *Journal of Computational Physics* 390, 548-594.

Nangia, N., Patankar, N.A., Bhalla, A.P.S., 2019b. A DLM immersed boundary method based wave-structure interaction solver for high density ratio multiphase flows. *Journal of Computational Physics* 398, 108804.

Nguyen, C.T., Oterkus, S., 2019. Peridynamics for the thermomechanical behavior of shell structures. *Engineering Fracture Mechanics* 219, 106623.

Oterkus, E., Madenci, E., 2011. Peridynamic Theory for Damage Initiation and Growth in Composite Laminate. *Key Engineering Materials* 488-489, 355-358.

Oterkus, E., Madenci, E., 2012. Peridynamic analysis of fiber-reinforced composite materials. *Journal of Mechanics of Materials and Structures* 7 (1), 45-84.

Oterkus, E., Madenci, E., Weckner, O., Silling, S., Bogert, P., Tessler, A., 2012. Combined finite element and peridynamic analyses for predicting failure in a stiffened composite curved panel with a central slot. *Composite Structures* 94 (3), 839-850.

Oterkus, S., 2015. *Peridynamics for the solution of multiphysics problems*. The University of Arizona, Arizona.

Oterkus, S., Fox, J., Madenci, E., 2013. Simulation of electro-migration through peridynamics, *Electronic Components and Technology Conference (ECTC), 2013 IEEE 63rd*. IEEE, pp. 1488-1493.

Oterkus, S., Madenci, E., 2015. Peridynamics for Antiplane Shear and Torsional Deformations. *Journal of Mechanics of Materials and Structures* 10 (2), 167-193.

Oterkus, S., Madenci, E., 2017. Peridynamic modeling of fuel pellet cracking. *Engineering Fracture Mechanics* 176, 23-37.

Oterkus, S., Madenci, E., Agwai, A., 2014a. Fully coupled peridynamic thermomechanics. *Journal of the Mechanics and Physics of Solids* 64, 1-23.

Oterkus, S., Madenci, E., Agwai, A., 2014b. Peridynamic thermal diffusion. *Journal of Computational Physics* 265, 71-96.

Oterkus, S., Madenci, E., Oterkus, E., 2017. Fully coupled poroelastic peridynamic formulation for fluid-filled fractures. *Engineering Geology* 225, 19-28.

Oterkus, S., Madenci, E., Oterkus, E., Hwang, Y., Bae, J., Han, S., 2014c. Hygro-thermo-mechanical analysis and failure prediction in electronic packages by using peridynamics, *2014 IEEE 64th Electronic Components and Technology Conference (ECTC)*, pp. 973-982.

- Pahar, G., Dhar, A., 2016. Mixed miscible-immiscible fluid flow modelling with incompressible SPH framework. *Engineering Analysis with Boundary Elements* 73, 50-60.
- Patel, J.K., Natarajan, G., 2017. A novel consistent and well-balanced algorithm for simulations of multiphase flows on unstructured grids. *Journal of Computational Physics* 350, 207-236.
- Remington, B.A., Weber, S.V., Marinak, M.M., Haan, S.W., Kilkenny, J.D., Wallace, R.J., Dimonte, G., 1995. Single - mode and multimode Rayleigh - Taylor experiments on Nova. *Physics of Plasmas* 2 (1), 241-255.
- Ren, H., Zhuang, X., 2018. *A Nonlocal Formulation for Weakly Compressible Fluid*. Springer Singapore, Singapore, pp. 835-850.
- Rezavand, M., Taeibi-Rahni, M., Rauch, W., 2018. An ISPH scheme for numerical simulation of multiphase flows with complex interfaces and high density ratios. *Computers & Mathematics with Applications* 75 (8), 2658-2677.
- Shadloo, M.S., Zainali, A., Yildiz, M., 2012. Simulation of single mode Rayleigh–Taylor instability by SPH method. *Computational Mechanics* 51 (5), 699-715.
- Shojaei, A., Galvanetto, U., Rabczuk, T., Jenabi, A., Zaccariotto, M., 2019. A generalized finite difference method based on the Peridynamic differential operator for the solution of problems in bounded and unbounded domains. *Computer Methods in Applied Mechanics and Engineering* 343, 100-126.
- Silling, S.A., 2000. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids* 48 (1), 175-209.
- Silling, S.A., 2010. Linearized Theory of Peridynamic States. *Journal of Elasticity* 99 (1), 85-111.
- Silling, S.A., Parks, M.L., Kamm, J.R., Weckner, O., Rassaian, M., 2017. Modeling shockwaves and impact phenomena with Eulerian peridynamics. *International Journal of Impact Engineering* 107, 47-57.
- Sloper, K.S., Dourmashkin, R.R., Bird, R.B., Slavin, G., Webster, A.D.B., 1982. Chronic Malabsorption Due to Cryptosporidiosis in a Child with Immunoglobulin Deficiency. *Gut* 23 (1), 80-82.
- Suresh, P., Kumar, S.P., Pantaik, B., 2019a. A Comparative Study of Two Different Density Estimation Techniques for Multi-Phase Flow Simulations Using SPH. *International Journal for Computational Methods in Engineering Science and Mechanics* 20 (1), 29-47.
- Suresh, P., Kumar, S.S.P., Pantaik, B.S.V., 2019b. A Comparative Study of Two Different Density Estimation Techniques for Multi-Phase Flow Simulations Using SPH. *International Journal for Computational Methods in Engineering Science and Mechanics* 20 (1), 29-47.
- Talon, L., Meiburg, E., 2011. Plane Poiseuille flow of miscible layers with different viscosities: instabilities in the Stokes flow regime. *Journal of Fluid Mechanics* 686, 484-506.
- Tartakovsky, A.M., Meakin, P., 2005. A smoothed particle hydrodynamics model for miscible flow in three-dimensional fractures and the two-dimensional Rayleigh–Taylor instability. *Journal of Computational Physics* 207 (2), 610-624.
- Tessler, A., Di Sciuva, M., Gherlone, M., 2009. A Refined Zigzag Beam Theory for Composite and Sandwich Beams. *Journal of Composite Materials* 43 (9), 1051-1081.
- TingYe, DingyiPan, CanHuang, MoubinLiu, 2019. Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications. *Physics of Fluids* 31 (1), 011301.
- Tu, Q., Li, S., 2017. An updated Lagrangian particle hydrodynamics (ULPH) for Newtonian fluids. *Journal of Computational Physics* 348, 493-513.
- Turns, S.R., 1996. *An introduction to combustion*. McGraw-hill New York.

- Vacondio, R., Rogers, B.D., Stansby, P.K., Mignosa, P., Feldman, J., 2013. Variable resolution for SPH: A dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering* 256, 132-148.
- van der Pijl, S.P., Segal, A., Vuik, C., Wesseling, P., 2005. A mass-conserving Level-Set method for modelling of multi-phase flows. *International Journal for Numerical Methods in Fluids* 47 (4), 339-361.
- Wang, H., Oterkus, E., Oterkus, S., 2018a. Peridynamic modelling of fracture in marine lithium-ion batteries. *Ocean Engineering* 151, 257-267.
- Wang, H., Oterkus, E., Oterkus, S., 2018b. Predicting fracture evolution during lithiation process using peridynamics. *Engineering Fracture Mechanics* 192, 176-191.
- Wang, H., Oterkus, E., Oterkus, S., 2018c. Three-Dimensional Peridynamic Model for Predicting Fracture Evolution during the Lithiation Process. *Energies* 11 (6), 1461.
- Wang, J., Zhang, X., 2019. Improved Moving Particle Semi-implicit method for multiphase flow with discontinuity. *Computer Methods in Applied Mechanics and Engineering* 346, 312-331.
- Wang, Z.-B., Chen, R., Wang, H., Liao, Q., Zhu, X., Li, S.-Z., 2016. An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modelling* 40 (23-24), 9625-9655.
- Xu, R., Stansby, P., Laurence, D., 2009. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics* 228 (18), 6703-6725.
- Yan, J., Li, S., Zhang, A.M., Kan, X., Sun, P.-N., 2019. Updated Lagrangian Particle Hydrodynamics (ULPH) modeling and simulation of multiphase flows. *Journal of Computational Physics* 393, 406-437.
- Yang, Q., Yao, J., Huang, Z., Asif, M., 2019. A comprehensive SPH model for three-dimensional multiphase interface simulation. *Computers & Fluids* 187, 98-106.
- Zhang, A., Sun, P., Ming, F., 2015. An SPH modeling of bubble rising and coalescing in three dimensions. *Computer Methods in Applied Mechanics and Engineering* 294, 189-209.
- Zhang, Z.L., Walayat, K., Huang, C., Chang, J.Z., Liu, M.B., 2019. A finite particle method with particle shifting technique for modeling particulate flows with thermal convection. *International Journal of Heat and Mass Transfer* 128, 1245-1262.

Appendix A Derivation of non-local derivatives for a vector function

In this appendix, the derivation of the PDDO formulation for the derivatives of a vector function is provided. Since only the two-dimensional fluid flow is considered, and the highest differential order in the governing equation is the second order, the PDDO for a two-dimensional problem with up to second-order derivatives is discussed here. Then, the formula of the first and second-order derivatives of a vector function in an integral form is provided.

The peridynamic differential operator is constructed based on the Taylor series expansion and the orthogonal functions (Madenci et al., 2016). For a scalar function in a 2D space, $f(\mathbf{x}) = f(x_1, x_2)$, the Taylor series expansion is expressed as

$$\begin{aligned} f(\mathbf{x} + \boldsymbol{\xi}) = & f(\mathbf{x}) + \xi_1 \frac{\partial f(\mathbf{x})}{\partial x_1} + \xi_2 \frac{\partial f(\mathbf{x})}{\partial x_2} \\ & + \frac{1}{2} \xi_1^2 \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} + \frac{1}{2} \xi_2^2 \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} + \xi_1 \xi_2 \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} + R_{\mathbf{x},2}(\boldsymbol{\xi}) \end{aligned} \quad (\text{A.1})$$

where $R_{\mathbf{x},2}(\boldsymbol{\xi})$ is the remainder term which is neglected in the derivation. The difference between $f(\mathbf{x} + \boldsymbol{\xi})$ and $f(\mathbf{x})$ consequently can be expressed in a compact form as

$$f(\mathbf{x} + \boldsymbol{\xi}) - f(\mathbf{x}) = \sum_{n_1=0}^2 \sum_{n_2=0}^{2-n_1} \frac{1}{n_1! n_2!} \xi_1^{n_1} \xi_2^{n_2} \frac{\partial^{n_1+n_2} f(\mathbf{x})}{\partial x_1^{n_1} \partial x_2^{n_2}}; 0 < n_1 + n_2 \leq 2 \quad (\text{A.2})$$

with n_1 and n_2 representing the differential order of $f(\mathbf{x})$ with respect to x_1 and x_2 .

The term $g^{p_1 p_2}(\boldsymbol{\xi})$ represents the PD operator corresponding to the p_1 order and p_2 order derivatives of the function to x_1 and x_2 . Then multiplying the PDDO $g^{p_1 p_2}(\boldsymbol{\xi})$ on both sides of Eq. (A.2) and integrating over the neighbourhood $H_{\mathbf{x}}$, Eq. (A.2) results in

$$\begin{aligned} & \int_{H_{\mathbf{x}}} (f(\mathbf{x} + \boldsymbol{\xi}) - f(\mathbf{x})) g^{p_1 p_2}(\boldsymbol{\xi}) dV' \\ & = \sum_{n_1=0}^2 \sum_{n_2=0}^{2-n_1} \frac{1}{n_1! n_2!} \int_{H_{\mathbf{x}}} \frac{\partial^{n_1+n_2} f(\mathbf{x})}{\partial x_1^{n_1} \partial x_2^{n_2}} \xi_1^{n_1} \xi_2^{n_2} g^{p_1 p_2}(\boldsymbol{\xi}) dV' \end{aligned} \quad (\text{A.3})$$

The peridynamic differential operator, $g^{p_1 p_2}(\boldsymbol{\xi})$, is predefined to have the orthogonal property as

$$\int_{H_{\mathbf{x}}} \xi_1^{n_1} \xi_2^{n_2} g^{p_1 p_2}(\boldsymbol{\xi}) dV' = n_1! n_2! \delta_{n_1 p_1} \delta_{n_2 p_2} \quad (\text{A.4})$$

where $\delta_{n_1 p_1}$ and $\delta_{n_2 p_2}$ are the Kronecker delta functions. Therefore, Eq.(A.3) can be further calculated by incorporating Eq.(A.4), leading to

$$\int_{H_x} (f(\mathbf{x} + \xi) - f(\mathbf{x})) g^{p_1 p_2}(\xi) dV' = \frac{\partial^{p_1 + p_2} f(\mathbf{x})}{\partial x_1^{p_1} \partial x_2^{p_2}} \quad (\text{A.5})$$

The above content describes the PDDO application on the scalar function differential converting. The following content will briefly describe the application on the vector functions.

For a vector function, $\mathbf{f}(\mathbf{x}) = \{f_1(x_1, x_2), f_2(x_1, x_2)\}$, its divergence is defined as

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = \frac{\partial f_1(x_1, x_2)}{\partial x_1} + \frac{\partial f_2(x_1, x_2)}{\partial x_2} \quad (\text{A.6})$$

By applying Eq.(A.5), the differentials can be calculated as

$$\frac{\partial f_1(\mathbf{x})}{\partial x_1} = \int_{H_x} (f_1(\mathbf{x} + \xi) - f_1(\mathbf{x})) g^{10}(\xi) dV' \quad (\text{A.7a})$$

$$\frac{\partial f_2(\mathbf{x})}{\partial x_2} = \int_{H_x} (f_2(\mathbf{x} + \xi) - f_2(\mathbf{x})) g^{01}(\xi) dV' \quad (\text{A.7b})$$

Therefore, the divergence can be expressed as

$$\begin{aligned} \nabla \cdot \mathbf{f}(\mathbf{x}) &= \int_{H_x} [(f_1(\mathbf{x} + \xi) - f_1(\mathbf{x})) g^{10}(\xi) + (f_2(\mathbf{x} + \xi) - f_2(\mathbf{x})) g^{01}(\xi)] dV' \\ &= \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV' \end{aligned} \quad (\text{A.8})$$

The gradient of $\mathbf{f}(\mathbf{x})$ in a classical form is

$$\nabla \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1(x_1, x_2)}{\partial x_1} & \frac{\partial f_1(x_1, x_2)}{\partial x_2} \\ \frac{\partial f_2(x_1, x_2)}{\partial x_1} & \frac{\partial f_2(x_1, x_2)}{\partial x_2} \end{bmatrix} \quad (\text{A.9})$$

In a similar process, applying the PDDO for the first order, $\mathbf{g}_1(\xi)$, the right-hand side of Eq.(A.9) will become

$$\begin{aligned}
\begin{bmatrix} \frac{\partial f_1(x_1, x_2)}{\partial x_1} & \frac{\partial f_1(x_1, x_2)}{\partial x_2} \\ \frac{\partial f_2(x_1, x_2)}{\partial x_1} & \frac{\partial f_2(x_1, x_2)}{\partial x_2} \end{bmatrix} &= \int_{H_x} \begin{bmatrix} (f_1(\mathbf{x} + \xi) - f_1(\mathbf{x}))g^{10}(\xi) & (f_1(\mathbf{x} + \xi) - f_1(\mathbf{x}))g^{10}(\xi) \\ (f_2(\mathbf{x} + \xi) - f_2(\mathbf{x}))g^{10}(\xi) & (f_2(\mathbf{x} + \xi) - f_2(\mathbf{x}))g^{01}(\xi) \end{bmatrix} dV' \\
&= \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \otimes \mathbf{g}_1(\xi) dV'
\end{aligned} \tag{A.10}$$

Finally, the integral form of the gradient is

$$\nabla \mathbf{f}(\mathbf{x}) = \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \otimes \mathbf{g}_1(\xi) dV' \tag{A.11}$$

As to the Laplacian operator, its original formulation is

$$\Delta \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f_1(x_1, x_2)}{\partial x_1^2} + \frac{\partial^2 f_1(x_1, x_2)}{\partial x_2^2} \\ \frac{\partial^2 f_2(x_1, x_2)}{\partial x_1^2} + \frac{\partial^2 f_2(x_1, x_2)}{\partial x_2^2} \end{bmatrix} \tag{A.12}$$

Applying the PDDO matrix $\mathbf{g}_2(\xi)$ for second-order derivatives as

$$\begin{aligned}
\begin{bmatrix} \frac{\partial^2 f_1(x_1, x_2)}{\partial x_1^2} + \frac{\partial^2 f_1(x_1, x_2)}{\partial x_2^2} \\ \frac{\partial^2 f_2(x_1, x_2)}{\partial x_1^2} + \frac{\partial^2 f_2(x_1, x_2)}{\partial x_2^2} \end{bmatrix} &= \int_{H_x} \begin{bmatrix} (f_1(\mathbf{x} + \xi) - f_1(\mathbf{x}))(g^{20}(\xi) + g^{02}(\xi)) \\ (f_2(\mathbf{x} + \xi) - f_2(\mathbf{x}))(g^{20}(\xi) + g^{02}(\xi)) \end{bmatrix} dV' \\
&= \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \text{Tr}(\mathbf{g}_2(\xi)) dV'
\end{aligned} \tag{A.13}$$

As a result, the Laplacian of the vector function can be expressed in an integral formulation as

$$\Delta \mathbf{f}(\mathbf{x}) = \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \text{Tr}(\mathbf{g}_2(\xi)) dV' \tag{A.14}$$

In conclusion, the derivation process of Eq.(10a) in Section 2 is explained here as

$$\begin{aligned}
\nabla \cdot \mathbf{f}(\mathbf{x}) &= \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \cdot \mathbf{g}_1(\xi) dV'; \\
\nabla \mathbf{f}(\mathbf{x}) &= \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \otimes \mathbf{g}_1(\xi) dV'; \\
\Delta \mathbf{f}(\mathbf{x}) &= \int_{H_x} (\mathbf{f}(\mathbf{x} + \xi) - \mathbf{f}(\mathbf{x})) \text{Tr}(\mathbf{g}_2(\xi)) dV';
\end{aligned} \tag{A.15}$$

The peridynamic differential operator, g for 2D problems can be constructed as a polynomial function as (Madenci et al., 2016)

$$\text{for } g^{10}(\xi): g^{10}(\xi) = (a_{10}^{10})w_{\xi_1} + (a_{01}^{10})w_{\xi_2} + (a_{20}^{10})w_{\xi_1^2} + (a_{02}^{10})w_{\xi_2^2} + (a_{11}^{10})w_{\xi_1\xi_2} \quad (\text{A.16a})$$

$$\text{for } g^{01}(\xi): g^{01}(\xi) = (a_{10}^{01})w_{\xi_1} + (a_{01}^{01})w_{\xi_2} + (a_{20}^{01})w_{\xi_1^2} + (a_{02}^{01})w_{\xi_2^2} + (a_{11}^{01})w_{\xi_1\xi_2} \quad (\text{A.16b})$$

$$\text{for } g^{20}(\xi): g^{20}(\xi) = (a_{10}^{20})w_{\xi_1} + (a_{01}^{20})w_{\xi_2} + (a_{20}^{20})w_{\xi_1^2} + (a_{02}^{20})w_{\xi_2^2} + (a_{11}^{20})w_{\xi_1\xi_2} \quad (\text{A.16c})$$

$$\text{for } g^{02}(\xi): g^{02}(\xi) = (a_{10}^{02})w_{\xi_1} + (a_{01}^{02})w_{\xi_2} + (a_{20}^{02})w_{\xi_1^2} + (a_{02}^{02})w_{\xi_2^2} + (a_{11}^{02})w_{\xi_1\xi_2} \quad (\text{A.16d})$$

$$\text{for } g^{11}(\xi): g^{11}(\xi) = (a_{10}^{11})w_{\xi_1} + (a_{01}^{11})w_{\xi_2} + (a_{20}^{11})w_{\xi_1^2} + (a_{02}^{11})w_{\xi_2^2} + (a_{11}^{11})w_{\xi_1\xi_2} \quad (\text{A.16e})$$

which can also be expressed in a matrix form as (Madenci et al., 2016)

$$\begin{Bmatrix} g^{10}(\xi) \\ g^{01}(\xi) \\ g^{20}(\xi) \\ g^{02}(\xi) \\ g^{11}(\xi) \end{Bmatrix} = \begin{bmatrix} a_{10}^{10} & a_{01}^{10} & a_{20}^{10} & a_{02}^{10} & a_{11}^{10} \\ a_{10}^{01} & a_{01}^{01} & a_{20}^{01} & a_{02}^{01} & a_{11}^{01} \\ a_{10}^{20} & a_{01}^{20} & a_{20}^{20} & a_{02}^{20} & a_{11}^{20} \\ a_{10}^{02} & a_{01}^{02} & a_{20}^{02} & a_{02}^{02} & a_{11}^{02} \\ a_{10}^{11} & a_{01}^{11} & a_{20}^{11} & a_{02}^{11} & a_{11}^{11} \end{bmatrix} \begin{Bmatrix} w_{\xi_1} \\ w_{\xi_2} \\ w_{\xi_1^2} \\ w_{\xi_2^2} \\ w_{\xi_1\xi_2} \end{Bmatrix} \quad (\text{A.17})$$

and the weighted function is the same as Eq.(7). The term $a_{q_1q_2}^{p_1p_2}$ represents the coefficient of the polynomials with $0 < q_1 + q_2 \leq 2$. Then by substituting Eq.(A.16) into the orthogonal function Eq.(A.4), the following relationship can be obtained as

$$\mathbf{Aa} = \mathbf{b} \quad (\text{A.18})$$

with

$$\mathbf{A} = \int_{H_x} w \begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1^3 & \xi_1\xi_2^2 & \xi_1^2\xi_2 \\ \xi_1\xi_2 & \xi_2^2 & \xi_1^2\xi_2 & \xi_2^3 & \xi_1\xi_2^2 \\ \xi_1^3 & \xi_1^2\xi_2 & \xi_1^4 & \xi_1^2\xi_2^2 & \xi_1^3\xi_2 \\ \xi_1\xi_2^2 & \xi_2^3 & \xi_1^2\xi_2^2 & \xi_2^4 & \xi_1\xi_2^3 \\ \xi_1^2\xi_2 & \xi_1\xi_2^2 & \xi_1^3\xi_2 & \xi_1\xi_2^3 & \xi_1^2\xi_2^2 \end{bmatrix} dV' \quad (\text{A.19a})$$

and

$$\mathbf{b} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.19b})$$

$$\mathbf{a} = \begin{bmatrix} a_{10}^{10} & a_{01}^{10} & a_{20}^{10} & a_{02}^{10} & a_{11}^{10} \\ a_{10}^{01} & a_{01}^{01} & a_{20}^{01} & a_{02}^{01} & a_{11}^{01} \\ a_{10}^{20} & a_{01}^{20} & a_{20}^{20} & a_{02}^{20} & a_{11}^{20} \\ a_{10}^{02} & a_{01}^{02} & a_{20}^{02} & a_{02}^{02} & a_{11}^{02} \\ a_{10}^{11} & a_{01}^{11} & a_{20}^{11} & a_{02}^{11} & a_{11}^{11} \end{bmatrix}^T \quad (\text{A.19c})$$

Then according to Eq.(A.18) and Eq.(A.19), the following equation can be derived as

$$\begin{bmatrix} a_{10}^{10} & a_{01}^{10} & a_{20}^{10} & a_{02}^{10} & a_{11}^{10} \\ a_{10}^{01} & a_{01}^{01} & a_{20}^{01} & a_{02}^{01} & a_{11}^{01} \\ a_{10}^{20} & a_{01}^{20} & a_{20}^{20} & a_{02}^{20} & a_{11}^{20} \\ a_{10}^{02} & a_{01}^{02} & a_{20}^{02} & a_{02}^{02} & a_{11}^{02} \\ a_{10}^{11} & a_{01}^{11} & a_{20}^{11} & a_{02}^{11} & a_{11}^{11} \end{bmatrix} = \mathbf{a}^T = (\mathbf{A}^{-1}\mathbf{b})^T = \mathbf{A}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (\text{A.20})$$

Thus, Eq.(A.17) becomes

$$\begin{Bmatrix} g^{10}(\xi) \\ g^{01}(\xi) \\ g^{20}(\xi) \\ g^{02}(\xi) \\ g^{11}(\xi) \end{Bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{Bmatrix} w_{\xi_1}^\xi \\ w_{\xi_2}^\xi \\ w_{\xi_1}^{\xi^2} \\ w_{\xi_2}^{\xi^2} \\ w_{\xi_1\xi_2}^\xi \end{Bmatrix} \quad (\text{A.21})$$

The peridynamic differential operator for 2D problems can be calculated numerically as provided in Eq. (74) as an example.

Appendix B PST corrections for physical variables except for displacement

Regarding the PST, after the particle position being shifted, the fluid velocity \mathbf{v}_i , pressure p_i and density ρ_i for particle i , represented by ϕ_i , can be corrected according to the Taylor series expansion. Originally, the Taylor series expansion used in the PST (Xu et al., 2009) only has second-order accuracy. In the present work, the second-order derivatives are already calculated by using PPDO. Therefore, Taylor series expansion with a third-order error term can easily be adopted as (Heck, 1993)

$$\left(\phi_i^{n+1}\right)^{corrected} = \phi_i^{n+1} + (\nabla \phi)_i^n \cdot (\delta \mathbf{u}_i)^{PST} + \frac{1}{2} \left((\delta \mathbf{u}_i)^{PST} \right)^T \mathbf{H}_i (\delta \mathbf{u}_i)^{PST} + o\left(\left| (\delta \mathbf{u}_i)^{PST} \right|^3\right) \quad (\text{B.1})$$

where \mathbf{H}_i represents the Hessian matrix as (Heck, 1993)

$$\mathbf{H}_i = \begin{bmatrix} \frac{\partial^2 \phi_i}{\partial x_1^2} & \frac{\partial^2 \phi_i}{\partial x_1 \partial x_2} \\ \frac{\partial^2 \phi_i}{\partial x_1 \partial x_2} & \frac{\partial^2 \phi_i}{\partial x_2^2} \end{bmatrix} \quad (\text{B.2})$$

Consequently, the non-local form of the Hessian matrix can be derived as

$$\begin{aligned} \mathbf{H}_i &= \begin{bmatrix} \frac{\partial^2 \phi_i}{\partial x_1^2} & \frac{\partial^2 \phi_i}{\partial x_1 \partial x_2} \\ \frac{\partial^2 \phi_i}{\partial x_1 \partial x_2} & \frac{\partial^2 \phi_i}{\partial x_2^2} \end{bmatrix} = \sum_{j=1}^{N_i} \left(\phi(\mathbf{x}_j^n) - \phi(\mathbf{x}_i^n) \right) \begin{bmatrix} \mathbf{g}^{20}(\xi_{ij}^n) & \mathbf{g}^{11}(\xi_{ij}^n) \\ \mathbf{g}^{11}(\xi_{ij}^n) & \mathbf{g}^{02}(\xi_{ij}^n) \end{bmatrix} V_j \\ &= \sum_{j=1}^{N_i} \left(\phi(\mathbf{x}_j^n) - \phi(\mathbf{x}_i^n) \right) \mathbf{g}_2(\xi_{ij}^n) V_j \end{aligned} \quad (\text{B.3})$$

The PPDO form of Eq.(B.1) can be derived by replacing the PDDO gradient operator provided in Eq.(A.15) and PDDO form of the Hessian matrix in Eq.(B.3), resulting in

$$\begin{aligned} \left(\phi_i^{n+1}\right)^{corrected} &= \phi_i^{n+1} + \left[\sum_{j=1}^{N_i} \left(\phi(\mathbf{x}_j^n) - \phi(\mathbf{x}_i^n) \right) \mathbf{g}_1(\xi_{ij}^n) V_j \right] \cdot (\delta \mathbf{u}_i)^{PST} \\ &+ \frac{1}{2} \left((\delta \mathbf{u}_i)^{PST} \right)^T \left[\sum_{j=1}^{N_i} \left(\phi(\mathbf{x}_j^n) - \phi(\mathbf{x}_i^n) \right) \mathbf{g}_2(\xi_{ij}^n) V_j \right] (\delta \mathbf{u}_i)^{PST} \end{aligned} \quad (\text{B.4})$$

where the expression of $(\delta \mathbf{u}_i)^{PST}$ is provided in Eq.(59).

Therefore, the corrected velocity components become

$$\begin{aligned} (v_{i,x}^{n+1})^{corrected} &= v_{i,x}^{n+1} + \left[\sum_{j=1}^{N_i} (v_x(\mathbf{x}_j^n) - v_x(\mathbf{x}_i^n)) \mathbf{g}_1(\xi_{ij}^n) V_j \right] \cdot (\delta \mathbf{u}_i)^{PST} \\ &+ \frac{1}{2} \left((\delta \mathbf{u}_i)^{PST} \right)^T \left[\sum_{j=1}^{N_i} (v_x(\mathbf{x}_j^n) - v_x(\mathbf{x}_i^n)) \mathbf{g}_2(\xi_{ij}^n) V_j \right] (\delta \mathbf{u}_i)^{PST} \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned} (v_{i,y}^{n+1})^{corrected} &= v_{i,y}^{n+1} + \left[\sum_{j=1}^{N_i} (v_y(\mathbf{x}_j^n) - v_y(\mathbf{x}_i^n)) \mathbf{g}_1(\xi_{ij}^n) V_j \right] \cdot (\delta \mathbf{u}_i)^{PST} \\ &+ \frac{1}{2} \left((\delta \mathbf{u}_i)^{PST} \right)^T \left[\sum_{j=1}^{N_i} (v_y(\mathbf{x}_j^n) - v_y(\mathbf{x}_i^n)) \mathbf{g}_2(\xi_{ij}^n) V_j \right] (\delta \mathbf{u}_i)^{PST} \end{aligned} \quad (\text{B.6})$$

The corrected pressure values become

$$\begin{aligned} (P_i^{n+1})^{corrected} &= P_i^{n+1} + \left[\sum_{j=1}^{N_i} (P(\mathbf{x}_j^n) - P(\mathbf{x}_i^n)) \mathbf{g}_1(\xi_{ij}^n) V_j \right] \cdot (\delta \mathbf{u}_i)^{PST} \\ &+ \frac{1}{2} \left((\delta \mathbf{u}_i)^{PST} \right)^T \left[\sum_{j=1}^{N_i} (P(\mathbf{x}_j^n) - P(\mathbf{x}_i^n)) \mathbf{g}_2(\xi_{ij}^n) V_j \right] (\delta \mathbf{u}_i)^{PST} \end{aligned} \quad (\text{B.7})$$

It should be noted that the corrections for velocity components and pressure have limit effects on the accuracy improvement of the numerical results, less than 10% (Mokos et al., 2017). Therefore, the PST corrections for velocity and pressure provided from Eqs. (B.5) to (B.7) can be neglected for saving the computational time. Since in this study second-order PDDO equations are used, the higher-order terms in Eq. (B.1) can be easily adopted to increase the numerical accuracy of the PST method.

Furthermore, it should be noted that the PST applied here is only applicable to the fluid flow without free surface. If the free surface is involved, the improved version of PST by using Fick's law (Lind et al., 2012) should be adopted.