Research and application of SPH parallel algorithm based on particle decomposition

Siqi Wang *
School of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an 710054, China
*Corresponding author e-mail: 19208207029@stu.xust.edu.cn

Abstract. Smoothed Particle Hydrodynamics (SPH), as a typical meshless numerical method, has its advantage in modeling free surface flows. However, this method is computationally time-consuming. To resolve this problem, we propose a SPH parallel algorithm based on particle decomposition, which evenly distributes all particles to each process for calculation. In this circumstance, the functions including send, receive, and broadcast are called only once for each time step of communication, which is easy to implement and also has a good scalability. To verify the effectiveness of the proposed algorithm, we perform the numerical simulations of 2D dam-break flow and 3D droplet impact onto a liquid film. It is demonstrated that the proposed SPH parallel algorithm can significantly reduce the simulation time and has a massive advantage on 3D large-scale computation; the maximum speed-up ratio can up to 30 when the number of particles is more than million.

1. Introduction
The phenomenon of free surface flows exists widely in nature and industrial production, such as injection molding, hydraulic engineering, etc. The shape of the free surfaces in the flow process is complex, and it may produce many complex physical phenomena during the evolution, such as water splashing, water violent deformation near the boundary, rebound and integration with the water below, which is a strongly nonlinear complex problem. Therefore, it is of great theoretical value and practical significance to study how to simulate the flow process efficiently and accurately. The traditional grid-based numerical methods, for example finite difference method (FDM) and finite element method (FEM), need to apply extra interface tracking technology when solving the strongly nonlinear free surface flow problems, and the implementation procedure is complicated.

The Smoothed Particle Hydrodynamics (SPH) method is a Lagrangian meshless method, which can be employed for the simulations of free surface flows. This method was first proposed by Lucy [1] and Gingold [2] in 1977. Compared with the grid-based numerical methods, the SPH method is completely independent of the grid and has the advantages of Lagrangian characteristics, particle characteristics and adaptive characteristics, so it is very suitable for numerical simulation of complex interface problems such as large deformation and free surface flows. In 1994, Monaghan [3] applied SPH method to incompressible modeling of free surfaces for the first time, and constructed a simple computationally incompressible fluid model based on the compressibility assumption. In recent years, the SPH method has been successfully employed to solve problems such as incompressible flow [4, 5], multiphase flow
[6, 7], heat transfer [8, 9] and viscoelastic flow [10, 11]. It is worth noting that the SPH method is computationally heavy and time-consuming compared to the traditional grid-based methods, so it is necessary to parallelize the SPH program. At present, Cherfils et al. [12] designed a SPH parallel algorithm based on domain decomposition, and analyzed the parallel performance during the collapse of the 2D water column. Ihmsen et al. [13] presented a parallel framework for simulating fluids with SPH method and the presented experiments illustrated that the parallel framework can efficiently compute large numbers of time steps for large scenarios. Nishiura et al. [14] evaluated the effective computing performance and power efficiency of the SPH simulation on each architecture, and the GPU was found to produce the best arithmetic performance as a standalone device unit, and gave the most efficient power consumption. Fan et al. [15] applied the Open MP-based shared-memory parallel programming to the recently developed coupling method of state-based peridynamics and smoothed particle hydrodynamics (PD-SPH), and simulated dynamic soil fragmentation induced by the explosion of the buried explosives.

Based on the parallel programming platform of Message Passing Interface (MPI) and using C++ language as the programming language for algorithm implementation, this paper designs a SPH parallel algorithm based on particle decomposition. The algorithm distributes all particles to each process evenly for calculation, and calls only once the send, receive as well as broadcast function for each time step of communication, which is easy to implement and has a good scalability. The paper applied the algorithm to simulating 2D dam-break flow and 3D droplet impact onto a liquid film and analyzed the relationship among the number of processes, the number of particles, the parallel efficiency and the acceleration ratio. It is demonstrated that the number of particles is larger than one million, the maximum speed-up ratio can reach more than 30, which provides an efficient and accurate computing tool for the numerical simulation of 3D large-scale problems.

2. Mathematical model and method

2.1. Governing equations

In the Lagrangian coordinate system, the governing equations of 2D isothermal and Newtonian viscous fluid are

\[
\frac{1}{\rho} \frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{v} = 0
\]

\[
\frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \frac{\eta}{\rho} \nabla^2 \mathbf{v} + \mathbf{g}
\]

Where, \( \rho \), \( \mathbf{v} \) and \( p \) respectively represent fluid density, velocity and pressure, \( t \) is time, \( \eta \) is fluid viscosity, \( \mathbf{g} \) represents gravity, \( \frac{\partial}{\partial t} \) represents substance derivative, which is defined as

\[
\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla
\]

2.2. Smoothed particle hydrodynamics

The SPH method is one of the Lagrangian meshless particle methods. This method discretized the computational domain into a series of interacting and arbitrarily distributed particles, which can not only be used to estimate the field functions and approximate the governing equations, but also be used to characterize the physical properties of the medium system, such as density, velocity and pressure.

2.2.1. Numerical scheme. In the SPH method, there are many discrete forms of fluid control equations [16]. This paper discretizes the continuity equation and the momentum conservation equation into the following forms:

\[
\frac{D \rho}{Dt} = \sum_{j=1}^{N} m_j \mathbf{v}_j \cdot \frac{\partial W_j}{\partial \mathbf{x}_i}
\]
Where, \( m_j \) represents the mass of the particle \( j \), \( v_j, x_j \) represents the velocity and position of the particle \( j \), \( p_j \) is the pressure, \( \eta_j \) is the smoothing length defining the support domain of the kernel function. In this paper, the kernel function adopts the piecewise cubic spline function, and the radius of the influence domain is \( 2h \).

2.2.2. Boundary conditions. Boundary processing directly affects the efficiency and stability of numerical simulation, so it is very important for the entire calculation process of SPH. On the basis of previous work [17, 18], this paper proposes an enhanced boundary processing method composed of solid-wall particles and virtual particles outside the solid-wall boundary. First, a layer of solid-wall particles is arranged on the solid-wall boundary, and the spaces of particles are equal to the initial spaces of fluid particles. Different from the work of Monaghan [3], the solid-wall particles will no longer act as a repulsive force against the non-physical penetration of fluid particles. Similar to the method proposed by Morris et al. [19], the density and position of the solid-wall particle do not change during the whole calculation process, and the pressure can be calculated by the governing equation, which is the weighted average of the pressures of all fluid particles in its supporting domain:

\[
\rho_i = \frac{\sum_{j} (m_j / \rho_j) p_j W_{ij}}{\sum_{j} (m_j / \rho_j) W_{ij}}
\]

Where, \( P \) represents pressure, \( i \) represents boundary particle, and \( j \) represents fluid particle adjacent to particle \( i \).

Secondly, several layers of virtual particles are arranged outside the solid-wall boundary to make up for the lack of particles on the solid-wall boundary, and the spaces of particles are equal to the initial spaces of fluid particles. Similar to the solid wall particle, the density and position of the virtual particle outside the solid-wall do not change during the whole calculation process, and the pressure is the same as that of the connected solid wall particle. In order to meet the no-slip boundary condition, the velocities of both particles are set to zero. Figure 1. shows the connection between the virtual particles outside the solid-wall and the solid-wall particles.

\[
\frac{Dv_j}{Dt} = -\sum_{j=1}^{N} m_j \left( \frac{p_i + p_j}{\rho_j} \right) \frac{\partial W_{ij}}{\partial x_j} + \sum_{j=1}^{N} \left( \frac{p_i + p_j}{\rho_j} \right) \nabla \frac{\partial W_{ij}}{\partial x_j} \cdot v_j
\]

(5)
2.2.3. Equation of state. In the SPH method, incompressible fluid is usually regarded as weakly compressible fluid, that is, the variation range of fluid density is controlled within 1% by the equation of state to ensure that the fluid flow behavior is completely close to incompressible flow. The state equation used in this paper is as follows [3]:

$$p(\rho) = \frac{\rho_0 c^2}{\gamma} \left( \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right)$$  \hspace{1cm} (7)

Where, $\gamma$ is a constant, which is taken $\gamma = 7$ in most analyses; $\rho_0$ is the reference density, and its value is $\rho_0 = 1000\text{kg/m}^3$; $c$ represents the speed of sound, which usually is 10 times the maximum velocity of the fluid.

2.2.4. Time integral. Due to the low storage capacity and high computational efficiency of the leapfrog method, this paper selects the leapfrog method [15] for time integration of SPH discrete equations (4) and (5):

(a) After completing the first time step (t₀), the velocity and density of particle advance respectively by half a time step, while the displacement of particle advances by one time step [21]:

$$\begin{align*}
    t &= t_0 + \Delta t \\
    \mathbf{X}(t_0 + \Delta t/2) &= \mathbf{X}(t_0) + \frac{\Delta t}{2} \mathbf{F}(t_0) \\
    x_i^n(t_0 + \Delta t) &= x_i^n(t_0) + \Delta t \cdot v_i^n(t_0 + \Delta t/2)
\end{align*}$$  \hspace{1cm} (8)

Where, $\mathbf{X}_i$ represents the variable $\left( \rho_i, v_i^n, r_i^n \right)$, and $\mathbf{F}_i$ represents the summation part of the right end of the governing equation discretized by SPH.

(b) After that, before the start of each time step, the velocity and density of particle first advance by half a time step, keeping them consistent with the displacement, so as to achieve the consistency of calculation for each time step [21]:

$$\mathbf{X}(t) = \mathbf{X}(t - \Delta t/2) + \Delta t \frac{\Delta t}{2} \mathbf{F}(t - \Delta t)$$  \hspace{1cm} (9)

(c) The displacement, the velocity, and density of particle are advanced according to the standard leapfrog method at the end of each time step [21]:

$$\begin{align*}
    t &= t + \Delta t \\
    \mathbf{X}(t + \Delta t/2) &= \mathbf{X}(t - \Delta t/2) + \Delta t \mathbf{F}(t) \\
    x_i^n(t + \Delta t) &= x_i^n(t) + \Delta t v_i^n(t + \Delta t/2)
\end{align*}$$  \hspace{1cm} (10)

3. SPH parallel algorithm based on particle decomposition

3.1. Algorithm flow

Based on MPI parallel programming platform and using C++ as the programming language of algorithm implementation, this paper designs a set of SPH parallel algorithm based on particle decomposition. The basic idea of the algorithm is to calculate the interaction force between adjacent particles, assigning a certain number of particles according to the computing power of each processor, and carry out communication and parallel computation at each time step. Its main steps can be summarized as follows:

I. Preparation stage

Input the initial particle information and other data needed for calculation, and distribute all particles equally to each process: set the total number of particles as $N$, the total number of processes as $P$, and the number of processes be marked as $Z$ ($0 \leq Z \leq P - 1$). First calculate $N/P$ and $N\%P$, if $Z > N\%P$, then the starting and ending numbers of particles assigned to process $Z$ are respectively $Z \times N/P + N\%P$ and $(Z + 1) \times N/P + N\%P - 1$; Otherwise, the starting and ending numbers assigned to process $Z$ are respectively $Z \times (N/P + 1)$ and $(Z + 1) \times N/P + Z$.
II. Parallel calculations performed on each process
A. Calculate the pressures of boundary particles and virtual particles outside the solid-wall;
B. Send physical information, such as the positions, velocities and pressures of particles from process 1 to \( p-1 \), to process 0, which receives these information; Broadcast the physical information of particles from process 0 to other processes;
C. Use the linked list search method to search the neighboring particles of the particle, and calculate its kernel function and derivative function;
D. Solve the fluid control equation and calculate the density, velocity, pressure of particle and other physical change rates;
E. Carry out time integration to obtain the physical information of each particle at the next moment;
III. Serial post-processing analysis
Output particle information for post-processing analysis.

The advantage of SPH parallel algorithm based on particle decomposition is that each process is responsible for maintaining a fixed part of particles, regardless of the actual physical spaces of particles. In addition, the send, receive as well as broadcast function are called only once for each time step of communication, so the algorithm is easy to implement and has a good scalability.

3.2. Evaluate the parameters of parallel algorithms
Speed-up ratio and parallel efficiency are two key parameters to evaluate the performance of parallel algorithms [22]. Speed-up ratio refers to the ratio of serial running time to parallel running time of the same task,

\[
S_u = \frac{T_s}{T_p}
\]

Where, \( T_s \) represents the serial running time of the task, \( T_p \) represents the parallel running time of the task, and \( n \) represents the total number of processes used.

Parallel efficiency refers to the ratio of the parallel speed-up ratio to the total number of processes,

\[
E_p = \frac{S_u}{n}
\]

Generally \( E_p \) is less than or equal to 1, the closer to 1, the higher the parallel acceleration efficiency.

4. Parallel computing model
4.1. Validity verification
In order to verify the effectiveness of SPH method in simulating the free surface flow problem, a numerical simulation of the 2D dam-break initial state model, and its geometric dimensions are consistent with those in the literature [13, 23], that is, the height of dam-break water body is \( H = 1m \), the length is \( L = 2m \), and the width of flume is \( d = 5.366m \). The fluid density is \( \rho = 1000 \text{ kg} \cdot \text{m}^{-3} \), viscosity is \( \eta = 10^{-3} \text{ Pa} \cdot \text{s} \), gravity acceleration is \( g = 9.81 \text{ m} \cdot \text{s}^{-2} \), number of fluid particles used is \( N_f = 5000 \), time step is \( \Delta t = 1.0 \times 10^{-5} \text{ s} \).

![Figure 2. Sketch of 2D dam-break flow.](image)
Figure 3. shows the particle distribution of dam-break flow at different times, where \( t^* = t\sqrt{g/H} \) is the dimensionless time. It can be seen that under the action of gravity, the dam begins to collapse and propagates along the ground \((t^*=1.62)\), and then hits the right wall to form a vertical jet \((t^*=4.80)\). When the vertical jet reaches a certain height, due to energy dissipation and the action of gravity, it reverses backwards \((t^*=5.70)\), then drops, and finally fuses with the fluid below \((t^*=12.00)\). Obviously, the SPH method can accurately reproduce a variety of complex physical phenomena, such as water splashing, water violent deformation near the boundary, rebound and integration with the below water in the whole process of water column collapse, which show the advantages of SPH method in solving the problem of free surface flow.

Figure 4. Depicts the front position evolution computed, and this result is compared with the results in literature [13, 23]. It can be seen that the results in this paper are the same as those in the literature [13, 23], which verifies the effectiveness of SPH method.

Figure 3. SPH particle distribution of dam-break flow at different times.
4.1.1. Speed-up ratio and parallel efficiency analysis. Table 1. Parallel performance analysis of dam-break flow shows the parallel results of this example using different numbers of processes to run 40,000 time steps. It can be seen that when two or four processes are used, the speedup ratio and parallelism efficiency can be better, but when the number of processes used increases to eight, the speedup ratio and parallelism efficiency decrease to a certain extent. This is due to the fact that the number of particles used in the simulation is small. When the number of processes used increases, the number of particles allocated to each process decreases correspondently. As a result, the proportion of data required to communicate in the total data increases, and the communication volume increases, so the speed-up ratio and efficiency decrease.

In addition, in order to further analyze the parallel performance when the calculation scale increases, Figure 5. shows the parallel efficiency of dam-break flow when the number of particles is \( N_f = 5000 \) and 20000 respectively. It can be seen that when the number of fluid particles used increases to \( N_f = 20000 \), the parallelism efficiency when using 8 processes is significantly higher than that of \( N_f = 5000 \), and is above 85%. Therefore, according to the equal-efficiency scalability measurement method, the parallel SPH program in this paper has a good scalability.

**Table 1.** Parallel performance analysis of dam-break flow.

| number of processes | number of particles | total time (s) | Speedup ratio \( S_n \) | efficiency \( E_m \) |
|---------------------|---------------------|---------------|----------------------|------------------|
| 1                   |                     | 1588.53       | −                    | −                |
| 2                   | 100×50              | 831.82        | 1.91                 | 95.6%            |
| 4                   | 100×50              | 446.25        | 3.56                 | 88.9%            |
| 8                   | 100×50              | 255.80        | 6.21                 | 77.6%            |

**Figure 4.** The time history of the front position of dam-break flow.

**Figure 5.** Parallel efficiency of dam-break flow at different computational scales.
4.2. 3D droplet impacting liquid film problem
The phenomenon of droplet impact on liquid film is common in various engineering applications and nature such as jet cooling, micro-nano particle synthesis, material surface processing, raindrop splashing, etc., and this phenomenon is a very complex strongly nonlinear transient impact and incompressible free surface flow problem. Under different conditions, the droplet impact on the liquid film will produce various complex physical phenomena such as precipitation, spreading, "crown" water splash and liquid splash. Wang et al. [24] analyzed the relationship among liquid film thickness, fluid viscosity and critical Weber number through experiments. Xie et al. [25] used the semi-implicit method of moving particles to study and analyze the impact of droplet impact velocity and diameter on the flow in the process of 2D droplet impacting the liquid film. At present, the numerical simulation of droplet impact on liquid film is mostly confined to 2D space, while the 3D numerical simulation is closer to the reality.

In this paper, parallel SPH numerical simulation is carried out for the problem of 3D droplet impacting on liquid film and the calculation model is shown in Figure 6. Among them, the droplet and the liquid film use the same liquid, the density is $\rho = 1200 \text{ kg/m}^3$, the viscosity is $\eta = 0.022 \text{ Pa}\cdot\text{s}$. Droplet diameter is $D = 0.0042 \text{ m}$, droplet impact velocity is $V = 5.09 \text{ m/s}$. The liquid film length and width are respectively $L_x = L_y = 5D$, and the liquid film thickness is $H = 0.5D$. The liquid film length and width are respectively $L_x = L_y = 5D$, and the liquid film thickness is $H = 0.5D$. The liquid film length and width are respectively $L_x = L_y = 5D$, and the liquid film thickness is $H = 0.5D$. The initial spacing between particles is set to $\delta_0 = 0.000105 \text{ m}$, and the total number of particles is set to $N = 1071521$, among which the total number of fluid particles is 825,421, the total number of boundary particles is 58,801 and the total number of virtual particles outside the solid-wall is 187,299. The time step is taken as $\Delta t = 5.0 \times 10^{-7} \text{ s}$, to ensure numerical stability [26].

![Figure 6. Sketch of 3D droplet impinging on a liquid film.](image)

Figure 7. shows the SPH results of the 3D droplet impacting the liquid film at four different moments. It can be seen that, at the moment of $t = 0.25 \text{ ms}$, the droplet impacts the static liquid film at a certain initial velocity, and some particles of the droplet and some particles of the impacted liquid film fuse with each other and leap out of the surface of the liquid film, forming a thin sheet jet. At the moment of $t = 0.50 \text{ ms}$, more droplets continue to impact the liquid film, and the liquid film gradually expands and moves around along the solid wall, while the thin slice jet also keeps moving upward, forming a more obvious "crown" shaped spray. At the moment of $t = 1.00 \text{ ms}$, due to the action of inertial force, the droplets continue to move downward and finally fuse with the liquid film, while part of the particles that fall off from the edge of the "crown" spray form small droplets and finally produce splashing phenomenon. Obviously, the SPH parallel algorithm in this paper can vividly capture a variety of complex physical changes, such as "crown"-shaped water splashes and water splashes caused by droplets impacting on the liquid film.
Figure 7. SPH simulation of 3D droplet impinging on a liquid film.

Table 2. Parallel performance analysis of 3D droplet impinging liquid film.

| number of processes | number of particles | total time (s) | Speedup ratio $S_m$ |
|---------------------|---------------------|----------------|---------------------|
| 1                   | 1071521             | 180.9s         | --                  |
| 4                   |                     | 56.8s          | 3.18                |
| 16                  |                     | 16.1s          | 11.23               |
| 64                  |                     | 5.82s          | 31.08               |

5. Conclusion
In order to resolve the problem of computationally heavy and time-consuming in SPH method, this paper proposes a SPH parallel algorithm based on particle decomposition. The algorithm distributes all the particles equally to each process for calculation, and only calls the sending, receiving and broadcasting functions once for each time step communication, which is easy to implement and has a good scalability. The algorithm is applied to simulate 2D dam-break flow and 3D droplet impact onto a liquid film. The conclusions are as follows:

1) The numerical simulation results of 2D dam-break problem are in good agreement with the literature results, which verifies the effectiveness of SPH method to simulate free surface flow problem;

2) When the number of particles is small and the number of processes is large, the proportion of communication data in the total data increases, that is, the amount of communication increases, which leads to a certain degree of decline in speed-up and parallel efficiency;

3) The SPH method can realistically reproduce a variety of complex phenomena such as "crown" splashing and splashing. The maximum speed-up ratio can up to 30 when the number of particles is more than million.

Acknowledgments
This work is financially supported by the National Natural Science Foundation of China (No. 12071367).
References

[1] Lucy L B. A numerical approach to the testing of the fission hypothesis [J]. Astronomical Journal, 1977, 83(1): 1013-1024.

[2] Gingold R A, Monaghan J J. Smoothed particle hydrodynamics theory and application to non-spherical stars[J], Monthly Notices of the Royal Astronomical Society, 1977, 181(1): 375-389.

[3] Monaghan J J. Simulating free surface flows with SPH[J]. Journal of Computational Physics, 1994, 110(2): 399-406.

[4] Shao S, Lo E.Y.M. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface[J]. Advances in Water Resources, 2003, 26(7): 787-800.

[5] Khayyer A, Shimizu Y, Gotoh H, et al. A coupled incompressible SPH-Hamiltonian SPH solver for hydroelastic FSI corresponding to composite structures[J]. Applied Mathematical Modelling, 2021, 94: 242-271.

[6] Zhang M Y, Deng X L, Shen Z. A robust sharp interface method for SPH. Engineering Analysis with Boundary Elements, 2019, 106: 275-285.

[7] Meng Z F, Wang P P, Zhang A M, et al. A multiphase SPH model based on Roe’s approximate Riemann solver for hydraulic flows with complex interface [J]. Computer Methods in Applied Mechanics and Engineering, 2020, 365: 112999.

[8] Long T, Yang P, Liu M. A novel coupling approach of smoothed finite element method with SPH for thermal fluid structure interaction problems [J]. International Journal of Mechanical Sciences, 2020, 174: 105558.

[9] Li G Y, Ma X J, Zhang B W, et al. An integrated smoothed particle hydrodynamics method for numerical simulation of the droplet impacting with heat transfer [J]. Engineering Analysis with Boundary Elements, 2021, 124: 1-13.

[10] Xu X, Ouyang J, Liu Q, et al. SPH simulations of 2D transient viscoelastic flows using Brownian configuration fields[J]. Journal of Non-Newtonian Fluid Mechanics, 2014, 208: 59-71.

[11] Xu X, Yu P, A multiscale SPH method for simulating transient viscoelastic flows using bead-spring chain model[J]. Journal of Non-Newtonian Fluid Mechanics, 2016, 229: 27-42.

[12] Cherfils J.M, Pinon G, Rivoalen E. JOSEPHINE: A parallel SPH code for free surface flows [J]. Computer Physics Communications, 2012, 183: 1468-1480.

[13] Ihmsen M, Akinci N, Becker M, et al. A parallel SPH implementation on multi-core CPUs[C]/Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2011, 30(1): 99-112.

[14] Nishiura D, Furuichi M, Sakaguchi H. Computational performance of a smoothed particle hydrodynamics simulation for shared-memory parallel computing[J]. Computer Physics Communications, 2015, 194: 18-32.

[15] Fan H, Li S. Parallel peridynamics–SPH simulation of explosion induced soil fragmentation by using OpenMP [J]. Computational Particle Mechanics, 2017, 4(2): 199-211.

[16] Liu M.B, Liu G.R, Smoothed particle hydrodynamics (SPH): an overview and recent developments [J]. Archives of computational methods in engineering, 2010, 17(1): 25-76.

[17] Xu X, Yu P. Modeling and simulation of injection molding process of polymer melt by a robust SPH method [J]. Applied Mathematical Modelling, 2017, 48: 384-409.

[18] Xu X, Yu P. Extension of SPH to simulate non-isothermal free surface flows during the injection molding process[J]. Applied Mathematical Modelling, 2019, 73: 715-731.

[19] Morris J P, Fox P J, Zhu Y. Modeling low Reynolds number incompressible flows using SPH[J]. Journal of Computational Physics, 1997, 136(1): 214-226.

[20] Fang J N, Owens R G, Tacher L, et al. A numerical study of the SPH method for simulating transient viscoelastic free surface flows[J]. Journal of Non-Newtonian Fluid Mechanics, 2006, 139(1/2): 68-84.

[21] Tao Jiang, Jie Ouyang, Binxin Yang, Jinlian Ren. The SPH method for simulating a viscoelastic drop impact and spreading on an inclined plate[J]. Computational Mechanics, 2010, 45(6).
[22] Fa N H, Li S. Parallel peridynamics – SPH simulation of explosion induced soil fragmentation by using OpenMP[J]. Computational Particle Mechanics, 2016, 4(2):1-13.
[23] A. Colagrossi, M. Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics[J]. Journal of computational physics, 2003, 191: 448-475.
[24] Wang A.B, Chen Ch.Ch. Splashing impact of a single drop onto very thin liquid films [J]. Physics of Fluids, 2000, 12: 2155-2158.
[25] Xie H, Koshizuka S, Oka Y. Modeling of a single drop impact onto liquid film using particle method [J]. International Journal for Numerical Methods in Fluids, 2004, 45:1009-1023.
[26] Xu Xiaoyang, Jiang Yao-Lin, Yu Peng. SPH simulations of 3D dam-break flow against various forms of the obstacle: Toward an optimal design[J]. Ocean Engineering, 2021, 229.