

Research Article

An Improved Particle Number-Based Oil Spill Model Using Implicit Viscosity in Marine Simulator

Shaoyang Qiu ¹, Hongxiang Ren,¹ Haijiang Li ¹, Rui Tao,¹ and Yi Zhou²

¹Navigation College, Dalian Maritime University, Dalian 116026, China

²CNOOC Energy Technology & Service- Oil Production Services Co, Tianjin 300452, China

Correspondence should be addressed to Haijiang Li; hai_nav@163.com

Received 23 February 2021; Revised 6 April 2021; Accepted 27 May 2021; Published 7 June 2021

Academic Editor: Yannis Dimakopoulos

Copyright © 2021 Shaoyang Qiu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Improving the physical realism of oil spill scenes in marine simulators can further enhance the emergency response capabilities of officials in charge and crew members and help reduce losses caused by oil spill disasters. In order to uniformly simulate the spreading, drift, breakup, and merging of oil spills at sea, we propose an improved divergence-free position-based fluid (DFPBF) framework based on the particle number density model. In our DFPBF framework, the governing equations for oil spills and ocean are discretized by Lagrangian particles, and the incompressibility of oil spills and ocean is ensured by solving the divergence-free velocity constraint solver and constant density constraint solver. In order to stably simulate the fate and transport of oil spills with higher viscosity, we introduce an implicit viscosity solution scheme for our DFPBF framework. The implicit solver uses a splitting concept to decouple viscosity solve and adopts an implicit scheme to discretize the integration of viscous force. Moreover, our DFPBF framework can ensure a divergence-free velocity field before applying the implicit viscosity scheme, which avoids the undesired bulk viscosity effects. The simulation results show that our DFPBF framework can stably simulate oil spills of various viscosities, especially high-viscosity crude oils.

1. Introduction

Oil spills at sea are one of the most dangerous maritime disasters. Oil pollution has a serious hazard to the marine ecological environment and marine biological resources. Oil spill pollution accidents can cause vast stretches of ocean to become hypoxic, leading to the smothering of sedentary species, such as seaweeds and shellfish [1, 2]. The fishing and mariculture industries will also be seriously affected by oil spills. If the spilled oil is not cleaned up in time, it may also cause explosions and fires, causing more serious economic losses and casualties. Oil spills include releases of crude oil from tankers, offshore platforms, and drilling rigs. In this paper, we mainly focus on the ship-source oil spills. When an oil spill occurs, if the competent authority and ship's officers can respond effectively and successfully, the damage caused by oil spill will be significantly reduced. However, effective and successful emergency response capability is usually obtained from a lot of practical experience or

simulation training [3]. It is unrealistic to accumulate practical experience from actual oil spills because it will often bring huge losses and even dangers, and no mistakes are allowed. The advantage of marine simulator is that it can build virtual oil spill scenes and can simulate oil spill scenes under different weather and sea conditions according to user's requirements. There is no risk during the training process, and the emergency response capability of competent authority and officers can be trained through a large number of repeated simulation training.

The main study purpose of this paper is to physically model the oil spill scenes at sea and improve the realism of the oil spill scene in the marine simulator so that the marine simulator can play a more important role in the oil spill emergency response training. The fate and transport of oil spills in marine environment include spreading, drift, evaporation, dispersion, emulsification, dissolution, photo-oxidation, sedimentation and sinking, shoreline interaction, and biodegradation. During the early stages of a spill,

spreading, drift, evaporation, dispersion, emulsification, and dissolution are most important [4]. In this paper, we mainly focus on the physical movement process of oil spills in the early stage, that is, spreading and drift. The main theme of this paper is as follows:

- (1) In this paper, we introduce an improved DFPBF (divergence-free position-based fluid) framework to model oil spill scenarios. The DFPBF framework is the application of the concept of position-based dynamics in fluid simulation. The DFPBF framework is a meshless method that discretizes the problem domain into a series of fluid particles carrying physical quantities, which has great advantages in simulating the spreading and drift of oil spills. In order to simulate the coupled movement of the oil spills and the ocean surface, we propose an improved particle number density model. All particles in the support domain, including oil particles and ocean particles, have the same rest density, which can effectively handle the discontinuities of physical quantities at the interfaces.
- (2) The viscosity of the oil has a great effect on the spreading and drift. Usually, low-viscosity oils spread much faster than those with high viscosity. Moreover, high-viscosity oils are more resistant to deformation during spreading and drift. The viscosity of oil will be very high in low-temperature environment. Although the explicit viscosity solvers are usually more efficient, it requires a smaller time step to ensure stability when simulating higher viscosity oils, which will decrease the performance. In addition, when the viscosity of the oil exceeds a certain range, it is difficult to maintain stability even if the time step is small. Therefore, in order to stably simulate oils with different viscosities, we introduced an implicit viscosity solution scheme for the DFPBF framework. Compared with the explicit solvers, the implicit viscosity solver can stably simulate oil spills with very high viscosity without being limited by the time step. Before applying the implicit viscosity solver, our DFPBF framework ensures a divergence-free velocity field through the velocity-based constraints, which avoids the undesired bulk viscosity effects. The implicit viscosity solver uses the matrix-free conjugate gradient method and the Jacobi preconditioner to accelerate convergence, which significantly improves the computation efficiency.

2. Related Work

2.1. Spreading Models of Oil Spills. Once an oil spill occurs, the oil will start to spread. The spreading speed of the spilled oil is related to the viscosity. Refined products or light crudes with lower viscosity spread faster, and their thickness will gradually reduce as it spreads. In contrast, heavy crudes with high viscosity spread slowly and fragment into patches easily [4]. Oil spill spreading models mainly include regression models and numerical models. Early oil spill spreading

models are mainly regression models, which usually need to fit the experimental or observed data. Fay [5, 6] proposed a three-regime model for the spreading of oil spills. The oil spill spreads under the action of gravity, viscosity, and surface tension. On the basis of Fay's work, Mackay et al. [7] considered the effect of oil thickness. However, both Fay and Mackay thought that the oil spills were circular in shape, which is not consistent with the actual observation. Lehr et al. [8, 9] considered the effect of wind and Langmuir circulation on oil spreading and thought that the oil did not spread uniformly. The modified model proposed by Lehr alters the shape of oil from circle to ellipse with major axis in the direction of wind. Many researchers have improved the spreading of oil spills based on the Fay model, such as providing a surface area correction for the spreading formulas [10]. These regression models have been used to simulate oil spill scenarios in the marine simulator [11]. However, the regression models are difficult to simulate complex fluid-solid interaction scenes, such as the interaction scene between oil and booms.

Numerical models mainly refer to modeling oil spreading based on physical models, such as hydrodynamic methods [12, 13] and advection-diffusion equation [14, 15]. According to the different ways of model discretization, numerical methods can be divided into the Eulerian method [16, 17] and the Lagrangian method [14, 15]. The Eulerian method discretizes the problem domain into meshes, and the physical quantities at the vertices of the meshes change over time. The Eulerian method using volume fraction can easily trace the surface of the oil spill, but mesh generation for the problem domain is a prerequisite for the Eulerian method, and the accuracy is usually affected by the mesh resolution. The Lagrangian method is also called the meshless method, which discretizes the problem into a series of particles that carry physical quantities. The oil particles move according to the governing equations, and there is no need to generate meshes in the problem domain. The Lagrangian methods commonly used in oil spill simulation include the SPH (smoothed particle hydrodynamics) method [18–20] and the MPS (moving particle semi-implicit) method [21, 22]. The meshless method is suitable for simulating the spreading and dispersion of the oil, but it is difficult to track the free surface. Compared with regression models, numerical models have more advantages in simulating the effect of wind and waves on oil spreading and the interaction between oil and boom. On this basis, some researchers have proposed some hybrid methods that include both mesh and particles, such as the NBFLIP (narrow-band fluid implicit particle) method [23]. And, the NBFLIP method has been introduced into the marine simulator for oil spill simulation [24]. In recent years, with the development of artificial intelligence, some neural network models have gradually been used in prediction of oil transport, such as ANN (artificial neural network) [22]. Although these neural network models rely on the data generated by traditional models, the trained models have great application value in the oil spill scenarios of marine simulators.

2.2. Drift Models of Oil Spills. The drift of oil refers to the horizontal transport of oil spilled at sea under the action of ocean currents, waves, and winds. Hoult [25] derived the oil film drift equation, considering the effect of wind, waves, and tidal currents, which can be used to compute the drift velocity of any point on oil edge at different times and in different directions. Coppini et al. [26] adopted the Medsilk [27] drifting model for oil, and the spill is subdivided into a large number of Lagrangian particles. Coppini et al. considered the effect of wind field and turbulence on oil drift, and the information about oil slick features from remote-sensing observations was used to validate the oil drift model. The MIKE 21 SA model was used by De Padova et al. [28] to simulate the spatial evolution of floating oil. The Fokker-Planck equation for suspended oil particles is solved by the Lagrangian discrete parcel method. Dagestad et al. [29] developed an open-source Python-based framework for Lagrangian particle modeling. The framework is designed to be used for drift calculations in the ocean or atmosphere. OpenOil is a subclass of OpenDrift, and it is an oil drift model library including specific descriptions of processes relevant for oil spills. Based on the OpenOil, Röhrs et al. [30] simulated the advection of oil particles under the effect of currents, wind, and waves. It can be seen from the literatures that the Lagrangian particle method is one of the most widely used methods in oil spill drift modeling. With the development of technology and theory, some frameworks for simulating oil spill drift have been developed. However, most of the frameworks or models are used in oil spill forecasting, and there are relatively few applications in virtual reality oil spill simulation.

2.3. Viscosity Solvers for Meshless Frameworks. Viscosity is one of the important properties that affect the fate and transport of oil spills [31]. Considering the wide application of the meshless method in oil spill spreading and drift, we introduced a meshless DFPBF framework for unified modeling of oil spill spreading and drift. Therefore, in this section, we mainly investigate the viscosity solvers that can be used in meshless frameworks. Common viscosity solvers mainly include explicit viscosity solvers and implicit viscosity solvers. It is an intuitively explicit method to discretize the viscosity term by the Laplacian operator of the kernel function, but the second derivative changes the sign inside the kernel domain so that the normalization conditions are very poorly satisfied [32]. Müller et al. [33] designed a viscosity kernel function, which does not change the sign within the range of support. Brookshaw [34] derived the approximate form of the viscosity term based on the finite difference method and the gradient operator of the kernel function, which was later developed into a classic artificial viscosity by Monaghan [35, 36]. The artificial viscosity is Galilean invariant, vanishes for rigid body rotation, and conserves total linear and angular momenta. At the same time, Monaghan [36] proposed a concept of XSPH (“X” denotes the unknown factor). The velocity of the current particle is corrected by interpolating the velocity of the neighboring particles, which is a kind of visual viscosity. The

XSPH method has also been used for viscosity calculations in some works [37, 38]. The explicit viscosity solvers have fast calculating speed and are only suitable for simulating fluids with low viscosity.

In order to simulate high-viscosity fluids, some implicit solvers have been proposed. Takahashi et al. [39] derived the first derivative of strain rate tensor to model the implicit viscosity. However, the complex neighborhood search of this implicit scheme decreases the performance. Peer et al. [40] proposed a decomposition form of velocity gradient. They simulated the viscous behavior by modifying the traceless shear rate tensor obtained by decomposition. Bender and Koschier [41] defined a velocity constraint function on the basis of the strain rate tensor. This method adjusts the target strain rate tensor by changing the viscosity parameter to simulate fluids with different viscosities. Although the above implicit viscosity solvers can stably simulate fluids with high viscosity, they are computationally expensive. Therefore, these solvers cannot be directly introduced into the oil spill simulation in the marine simulator for the time being. On the basis of artificial viscosity in [35, 36], Weiler et al. [42] proposed an implicit scheme that does not depend on the strain rate tensor, which is currently the fastest implicit viscosity solver.

3. DFPBF Framework

In the DFPBF framework, the following Navier-Stokes equations are usually used as the governing equations of the ocean waves and oil spills:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0, \quad (1)$$

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\tau} + \mathbf{f}, \quad (2)$$

where ρ is the density, \mathbf{v} is the velocity, \mathbf{f} is the external body forces, and $\boldsymbol{\tau}$ is the stress tensor. Equation (1) is the continuity equation, and equation (2) is the momentum equation of viscous flow. For incompressible and isotropic Newtonian fluid, the stress tensor can be defined as

$$\boldsymbol{\tau} = -p\mathbf{I} + 2\mu\mathbf{E}, \quad (3)$$

where p is the pressure, \mathbf{I} is the identity tensor, μ is the dynamic viscosity, and \mathbf{E} is the strain rate tensor which is determined as

$$\mathbf{E} = \frac{1}{2}((\nabla\mathbf{v}) + (\nabla\mathbf{v})^T), \quad (4)$$

where $\nabla\mathbf{v}$ denotes the velocity gradient.

Inserting equations (3) and (4) into equation (2), we can obtain

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla \cdot (\nabla\mathbf{v}) + \mu \nabla \cdot (\nabla\mathbf{v})^T + \mathbf{f}. \quad (5)$$

According to the product rules of ∇ , $\nabla \cdot (\nabla\mathbf{v}) = \nabla^2\mathbf{v}$ and $\nabla \cdot (\nabla\mathbf{v})^T = \nabla(\nabla \cdot \mathbf{v})$. For incompressible fluids, the velocity divergence should be zero, that is, $\nabla \cdot \mathbf{v} = 0$. Therefore, the momentum equation can finally be expressed as

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{v} + \frac{\mathbf{f}}{\rho}, \quad (6)$$

where $\nu = \mu/\rho$ denotes the kinematic viscosity, $(1/\rho)\nabla p$ is the pressure term, and $\nu\nabla^2\mathbf{v}$ is the viscosity term.

3.1. Divergence-free Velocity Constraint. The divergence-free velocity field is one of the important conditions to ensure incompressibility and also an important prerequisite of momentum equation (6). The divergence-free velocity constraint solver is a velocity-based solver for continuity equation (1). The DFPBF framework enforces divergence-free velocity field by the following velocity divergence constraint:

$$C_i^v = \nabla \cdot \mathbf{v}_i = \sum_k \sum_{j=1}^{N_k} V_j \mathbf{v}_{ij} \cdot \nabla W_{ij} = 0, \quad (7)$$

where C_i^v is the velocity divergence constraint, V_j denotes the volume of neighboring particle j , $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, k is the index of the phase, and N_k is the number of neighboring particles in phase k .

The velocity divergence constraint aims to find a velocity correction $\Delta\mathbf{v}_i$ that can map the velocity field to the divergence-free field, that is, $C_i^v(\mathbf{v} + \Delta\mathbf{v}) = 0$. By applying Taylor expansion, we can obtain

$$C_i^v = C_i^v(\mathbf{v}) + \nabla_v C_i^v(\mathbf{v})\Delta\mathbf{v}_i, \quad (8)$$

where

$$\Delta\mathbf{v}_i = \lambda_i^v \nabla_v C_i^v(\mathbf{v}), \quad (9)$$

where λ_i^v is the Lagrange multiplier. Based on the concept of the gradient descent (GD) method, we enforce the velocity correction $\Delta\mathbf{v}_i$ along the direction of the gradient of the constraint. According to equation (7), the gradient of the velocity constraint $C_i^v(\mathbf{v})$ can be solved as

$$\nabla C_i^v(\mathbf{v}) = -\sum_k \sum_{j=1}^{N_k} V_j \nabla W_{ij}. \quad (10)$$

Similarly, according to Newton's third law, the gradient of the velocity constraint $\nabla C_{j\leftarrow i}^v(\mathbf{v})$ at the neighborhood particle \mathbf{x}_j can be defined as

$$\nabla C_{j\leftarrow i}^v(\mathbf{v}) = V_j \nabla W_{ij}. \quad (11)$$

Substituting equations (9)–(11) into equation (8), we can solve

$$\lambda_i^v = \frac{C_i^v(\mathbf{v})}{|\nabla C_i^v(\mathbf{v})|^2 + \sum_k \sum_{j=1}^{N_k} |\nabla C_{i\leftarrow j}^v(\mathbf{v})|^2} = C_i^v \alpha_i, \quad (12)$$

where $\nabla C_{i\leftarrow j}^v(\mathbf{v})$ and $\nabla C_{j\leftarrow i}^v(\mathbf{v})$ are equal in size and opposite in direction, and

$$\alpha_i = \frac{1}{|\sum_k \sum_{j=1}^{N_k} V_j \nabla W_{ij}|^2 + \sum_k \sum_{j=1}^{N_k} |V_j \nabla W_{ij}|^2 + \varepsilon}, \quad (13)$$

where ε is a term introduced to avoid singularities of the denominator.

The velocity correction of the particle i under the divergence-free velocity constraints can be derived as

$$\Delta\mathbf{v}_i = \lambda_i^v C_i^v(\mathbf{v}) + \sum_k \sum_{j=1}^{N_k} \lambda_j^v \nabla C_{i\leftarrow j}^v(\mathbf{v}) = \sum_k \sum_{j=1}^{N_k} \left(\lambda_i^v + \frac{\rho_{j,0}}{\rho_{i,0}} \lambda_j^v \right) V_j \nabla W_{ij}, \quad (14)$$

where $\rho_{i,0}$ and $\rho_{j,0}$ are the rest densities of the fluid phase containing particle i and j .

Algorithm 1 is the calculation flow of the divergence-free velocity constraint solver (Algorithm 1). η_v is the threshold of the density change rate, and *iteriter* is the number of iterations. In each iteration, firstly, the current velocity divergence constraint $C_i^v(\mathbf{v})$ is computed according to equation (7). The Lagrange multiplier λ_i^v for each particle is then computed according to equation (12). We can obtain the velocity correction $\Delta\mathbf{v}_i$ according to equation (14), and then update the particle velocity \mathbf{v}_i^* using $\Delta\mathbf{v}_i$ until the average density change rate $D\rho/Dt$ is less than the threshold η_v , that is, the constraint of divergence-free velocity field is satisfied.

3.2. Constant Density Constraint. In the meshless particle frameworks, there are two main ways to solve the pressure term in the momentum equation: EOS-based (equation of state) solvers and PPE-based (pressure Poisson equation) solvers. However, EOS-based solvers usually show weakly compressibility, while PPE is computationally expensive. The DFPBF framework adopts a more controllable way to solve the pressure term, that is, the concept of position-based dynamics (PBD) is introduced into the meshless framework. The PBD solver we proposed works at the position layer and directly solves a position correction according to the nonlinear density constraint so that the density remains constant. It has better incompressibility than the commonly used PPE solvers, and after our optimization, its calculation speed has been significantly improved.

In our optimized PBD solver, we use EOS-like nonlinear density constraint:

$$C_i = \frac{\rho_i}{\rho_{i,0}} - 1 = 0, \quad (15)$$

where C_i is the constraint function of the position of particle \mathbf{x}_i and its neighbors and ρ_i denotes the density of particle \mathbf{x}_i , and it is defined as

$$\rho_i = \rho_{i,0} \sum_k \sum_{j=1}^{N_k} V_j W_{ij}. \quad (16)$$

Equation (16) is an improved form of the particle number density model proposed by Solenthaler and Pajarola in [43]. Since we use a nonuniform sampling model in boundary handling, the volume of boundary particles will be adaptively adjusted with the shape of the rigid body, so we cannot directly compute the density of oil particles and ocean particles based on the particle number density [44]. Therefore, we modified the particle number density model and assumed that all particles in the neighborhood of i have only the rest density of particle i , as shown in Figure 1.

```

(1) while:  $((D\rho/Dt)_{\text{avg}} > \eta_v) \vee (\text{iter} < 1)$ :
(2)   for  $i$  in particles:
(3)     compute velocity constraint:  $C_i^v(\mathbf{v})$ 
(4)   for  $i$  in particles:
(5)     compute Lagrange multipliers:  $\lambda_i^v$  and  $\lambda_j^v$ 
(6)     compute velocity correction:  $\Delta\mathbf{v}_i$ 
(7)   for  $i$  in particles:
(8)     update velocity:  $\mathbf{v}^{\text{df}} = \mathbf{v}_i^* + \Delta\mathbf{v}_i$ 
    
```

ALGORITHM 1: Velocity divergence constraint (α, \mathbf{v}_i^*) (calculation flow of the divergence-free velocity constraint solver).

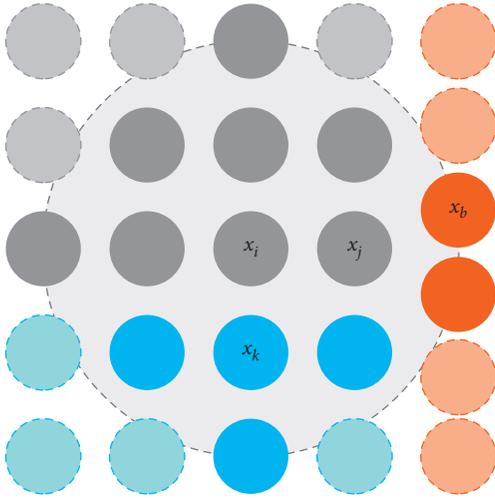


FIGURE 1: Schematic diagram of calculation of physical quantities at the interface. Blue particles \mathbf{x}_k denote the ocean, dark particles \mathbf{x}_i and \mathbf{x}_j denote the oil and orange particles, and \mathbf{x}_b denotes the nonuniform sampled rigid body boundaries. The big gray circle in the background is the support domain of oil particle \mathbf{x}_i .

Density constraint solver aims to find a position correction $\Delta\mathbf{x}$ for each particle, which is used to adjust the position distribution of fluid particles so that the density constraint $C_i(\mathbf{x} + \Delta\mathbf{x}) = 0$. Applying Taylor expansion, we can obtain

$$C_i = C_i(\mathbf{x}) + \nabla C_i(\mathbf{x})\Delta\mathbf{x}_i, \quad (17)$$

where the position correction $\Delta\mathbf{x}_i$ for particle i can be defined as

$$\Delta\mathbf{x}_i = \lambda_i \nabla C_i(\mathbf{x}), \quad (18)$$

where λ_i is the Lagrange multiplier for the density constraint solver. The particle position correction $\Delta\mathbf{x}_i$ is also enforced along the direction of the constraint gradient $\nabla C_i(\mathbf{x})$. It can ensure momentum conservation of the system. The constraint gradient at \mathbf{x}_i is

$$\nabla C_i(\mathbf{x}) = \sum_k \sum_{j=1}^{N_k} V_j \nabla W_{ij}. \quad (19)$$

The constraint gradient at \mathbf{x}_i is computed analogously to (11):

$$\nabla C_{j \leftarrow i}^v(\mathbf{x}) = -V_j \nabla W_{ij}. \quad (20)$$

Substituting equations (18)–(20) into equation (14), we can obtain

$$\lambda_i = C_i \alpha_i, \quad (21)$$

where α_i is defined in equation (13). In principle, the parameter α_i depends on the position distribution of the particles. When the position of the particle updates, the value of α_i also needs to be updated according to the latest particle position. However, repeated experiments show that when simulating oil spill scenarios, the value of α_i does not need to be updated in each iteration, which does not have a significant impact on the stability and convergence of the solver, but can significantly speed up the solution speed.

The position correction of the particle i under the constant density constraints can be derived as

$$\Delta\mathbf{x}_i = \lambda_i \nabla C_i(B) + \sum_k \sum_{j=1}^{N_k} \lambda_j \nabla C_{i \leftarrow j}(\mathbf{x}) \quad (22)$$

$$= \sum_k \sum_{j=1}^{N_k} \left(\lambda_i + \frac{\rho_{j,0}}{\rho_{i,0}} \lambda_j \right) V_j \nabla W_{ij}.$$

In fact, in the process of implementation, the density constraint usually uses inequality constraint, i.e., $C_i(\mathbf{x}) \leq 0$. This is because we have not considered the influence of air particles. Therefore, ocean particles, especially oil particles, at the free surface, usually have a lower density estimate due to the lack of neighboring particles. If the equality constraint in equation (15) is strictly implemented, it will cause particle clustering or clumping. The inequality constraint only adjusts the region where the particle density estimation is larger than rest density, which also improves the efficiency of the solver.

Algorithm 2 is the calculation flow of the constant density constraint solver (Algorithm 2). η_p is the threshold of density change and iteriter is the number of iterations. Since we adopted a pressure projection scheme to decouple the relationship between pressure term and velocity in momentum equation (2), at the beginning of each iteration, we need to first compute the intermediate density ρ_i^* based on the current particle position distribution. Then, we can compute the density constraint C_i and Lagrangian multiplier

```

(1) while:  $(\rho_{avg} - \rho_0 > \eta_\rho) \vee (\text{iter} < 2)$ 
(2)   for  $i$  in particles:
(3)     compute intermediate density:  $\rho_i^*$ 
(4)     compute density constraint:  $C_i(\mathbf{x})$ 
(5)     compute Lagrange multipliers:  $\lambda_i$  and  $\lambda_j$ 
(6)     compute position correction:  $\Delta \mathbf{x}_i$ 
(7)   for  $i$  in particles:
(8)     update position:  $\mathbf{x}^{cd} = \mathbf{x}_i^* + \Delta \mathbf{x}_i$ 

```

ALGORITHM 2: Constant density Constraint(α, \mathbf{v}_i^*) (calculation flow of the constant density constraint solver).

λ_i according to equations (15) and (21), respectively. Finally, we can compute a position correction $\Delta \mathbf{x}_i$ for each particle according to equation (22) and update the position of the particle using $\Delta \mathbf{x}_i$. Continue iterating until the average density change $\rho_{avg} - \rho_0$ is less than the threshold η_ρ .

When iteratively correct the particle position in Step 7, we introduce a relaxation scheme

$$\mathbf{x}^{cd} := \mathbf{x}_i^* + (1 + \omega)\Delta \mathbf{x}_i, \quad (23)$$

where ω is a user-defined relaxation factor and $\omega = 0.1$. The value of ω can be adjusted within a reasonable range, but should not exceed 1.0. In principle, the larger the value of ω , the faster the algorithm will converge, but it will also lead to incompressibility and stability issues. Since the parameter α_i is reused in this paper, a relatively smaller ω value should be selected to keep the algorithm stable. When the velocity is updated in the 8th step of the divergence-free velocity constraint solver shown in Algorithm 1, the relaxation factor ω can also be introduced to improve the convergence. However, experiments show that the velocity divergence solver only needs a few iterations to project the velocity field to a divergence-free field. Therefore, we only apply the relaxation scheme in the constant density constraint solver.

3.3. Implicit Viscosity Solver. Viscosity is one of the important properties that impact oil spreading and dispersion. Usually, low-viscosity oils flow more easily than those of higher viscosity. The viscosity of oils changes with their temperature and density. Generally, the viscosity of petroleum products is close to that of seawater, but the viscosity of crude oils is relatively high, and the viscosity will increase with the decrease of temperature. Therefore, the treatment of the viscosity term in the momentum equation (6) is very critical to the simulation of oil spill scenarios. At present, the most popular viscosity term solution schemes mainly include explicit schemes and implicit schemes.

3.3.1. Explicit Scheme. In the meshless method, we can directly discretize the viscosity term using particles:

$$\nabla^2 \mathbf{v}_i = \sum_j \frac{m_j}{\rho_j} \mathbf{v}_j \nabla^2 W_{ij}, \quad (24)$$

where $\nabla^2 W_{ij}$ denotes the Laplacian of kernel function. When computing the viscosity term, we only consider the

contribution of the neighboring particles in the same phase. However, in the common Gaussian kernel functions, the Laplacian operator $\nabla^2 W_{ij}$ will change the sign in the support domain, and the value is generally large, which will cause some nonphysical phenomena.

In order to solve the above problem, the first-order derivative of the kernel function can be combined with the finite difference method to discretize the viscosity term:

$$\nabla^2 \mathbf{v}_i = 2(d+2) \sum_j \frac{m_j}{\rho_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2 + 0.01h^2} \nabla W_{ij}, \quad (25)$$

where d is the spatial dimension, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, $|\mathbf{x}_{ij}|$ is the length of \mathbf{x}_{ij} , and $0.01h^2$ is introduced to avoid singularities. Equation (25) has the advantages of conservation of linear and angular momentum and is Galilean invariant. Since it is an explicit scheme, its calculation speed is faster. However, the stability of the explicit scheme is relatively poor and limited by the time step, and it is more suitable for simulating seawater with lower viscosity.

3.3.2. Implicit Scheme. In order to simulate higher viscosity oil spills, we introduced a stable implicit scheme. The second-order derivative of velocity $\nabla^2 \mathbf{v}_i$ can be obtained by computing the first-order derivative of the strain rate tensor \mathbf{E} in equation (4) [39]. Then, the first-order derivative $\nabla \mathbf{E}$ can be discretized by gradient operator ∇W_{ij} according to the concept of DFPBF framework. This method can stably simulate high-viscosity oil spills and is not limited by the time step size. However, the calculation of two first-order derivatives will lead to a sharp increase in the workload of neighborhood search, which greatly affects the performance. In order to ensure stable simulation and calculation efficiency at the same time, we introduce an implicit scheme based directly on equation (25) instead of using the strain rate. In our DFPBF framework, the viscosity term is solved after enforcing the divergence-free velocity field constraint:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}^{\text{df}} + \Delta t \frac{\mu}{\rho} \nabla^2 \mathbf{v}(t + \Delta t), \quad (26)$$

where \mathbf{v}^{df} denotes the divergence-free velocity field. Equation (26) is an implicit integration scheme and can be expressed as the following linear system:

$$(\mathbf{I} + \Delta t \mathbf{A}) \mathbf{v}(t + \Delta t) = \mathbf{v}^{\text{df}}, \quad (27)$$

where \mathbf{I} is the identity matrix and the matrix \mathbf{A} consists of 3×3 blocks:

$$\mathbf{A}_{ij} = -2(d+2) \frac{\mu \bar{m}_{ij}}{\rho_i \rho_j} \frac{\nabla W_{ij} \cdot \mathbf{x}_{ij}^T}{|\mathbf{x}_{ij}|^2 + 0.01h^2}, \quad (28)$$

$$\mathbf{A}_{ii} = -\sum_j \mathbf{A}_{ij},$$

where $\bar{m}_{ij} = (m_i + m_j)/2$ is the average mass to ensure a symmetric and positive definite matrix of coefficients. In order to improve efficiency and save memory consumption, the matrix-free conjugate gradient method is used to solve the linear system. In order to reduce the condition number of the coefficient matrix and improve the convergence, we adopt a preconditioning method. The block version of Jacobi preconditioner consists of the 3×3 blocks $I + \Delta t \mathbf{A}$.

When we couple the implicit viscosity solver into our DFPBF framework, we find that although implicit viscosity solver performs well in the simulation of high-viscosity fluid, it becomes unstable when the fluid particle velocity is high. In fact, under the same initial conditions, the velocity of low-viscosity fluid particles is higher than that of high-viscosity fluid particles. Moreover, the viscosity coefficient μ in the implicit viscosity solver is not consistent with the actual viscosity of fluid. In meshless frameworks, one of the purposes of introducing artificial viscosity is for the convergence and stability of numerical computation, and there is no strict correspondence with the actual physical meaning. For example, in the implicit viscosity solver, the motion of a fluid with a viscosity coefficient of $\mu = 10.0$ is almost the same as a fluid with an actual viscosity of 1.0×10^{-3} Pa s. In order to uniformly simulate fluids of different viscosities, we introduce a viscosity correction coefficient:

$$\mu^* = \gamma \mu, \quad (29)$$

where γ is a user-defined correction coefficient, and $\gamma = 1.0 \times 10^4$ in this paper.

3.4. Adaptive Time Step Size. In the simulation process, in order to speed up the physical calculation, it is usually preferred to choose a larger time step, but considering the stability, the time step size must also be small enough. Based on the above requirements, in meshless methods, the time step size can be estimated adaptively based on the CFL (Courant–Friedrichs–Lewy) condition. The adaptive scheme can maximize the time step while maintaining the stable convergence of the algorithm. Specifically, the time step should satisfy the following condition:

$$\Delta t_{\text{CFL}} \leq \lambda \frac{d}{|\mathbf{v}_{\text{max}}|}, \quad (30)$$

where d is the diameter of particle, \mathbf{v}_{max} is the maximum velocity of particle, and λ is a scaling factor, $\lambda = 0.4$. When $\lambda = 1.0$, the time step constraint in equation (29) means that the displacement of all particles in each time step should not exceed the particle diameter. On this basis, in order to prevent some particularly high-speed particles from affecting the performance of the algorithm, we specify the maximum and minimum time steps globally:

$$\Delta t = \max(t_{\text{min}}, \min(\Delta t_{\text{CFL}}, t_{\text{max}})), \quad (31)$$

where t_{min} and t_{max} denote the minimum and maximum thresholds of the time step, respectively.

Algorithm 3 shows the calculation flow of the DFPBF framework (Algorithm 3). We adopt a splitting scheme to decouple the solution of pressure, viscosity, and external forces. In each time step, we first compute the external forces and predict an intermediate velocity \mathbf{v}_i^* and intermediate position \mathbf{x}_i^* according to the external forces. After that, the adaptive time step Δt is computed to keep the simulation stable. Since the particle position has been updated, it is necessary to perform the neighborhood search again in Step 11. Since we have optimized the simulation loop, the parameter α_i^* is computed only before running the constraint solver. Adjust the particle distribution according to the constant density constraint solver in Algorithm 2. After applying the density constraint, a new position \mathbf{x}^{cd} is obtained, and then, the intermediate velocity \mathbf{v}_i^* can be updated according to \mathbf{x}^{cd} . We can get a divergence-free velocity field \mathbf{v}^{df} after applying the divergence-free velocity constraint solver in Algorithm 1. After obtaining the divergence-free velocity field, the viscosity solver is used to solve the viscosity term in the momentum equation (6).

4. Results and Discussion

In this section, we simulated some oil spill scenarios based on the oil spill model proposed in this paper. We use C++ to implement the physical calculation part of the oil spill model in Visual Studio 2019 and use OpenGL to visualize the oil spill scenarios. Our project runs on a workstation with an AMD Ryzen 7 3700X CPU (8-Core) and 32 GB RAM. The operating system is windows 10(x64). We adopt the adaptive single layer nonuniform boundary handling method, in which the velocity of static rigid boundary particles is 0.

Figure 2 is the schematic diagram of oil spill test scenario. In the experiment, the volume of ocean particles and oil particles are equal, and the particle radius is 0.015 L. The size of the bounding box is 4.0 L×4.0 L×2.0 L and the number of boundary particle is 175.1k. The size of ocean block is 4.0 L×0.3 L×2.0 L and the number of ocean particle is 78.5k. The size of oil block is 0.5 L×1.0 L×0.5 L and the number of oil particle is 8.2k. The oil block is located above the ocean block and gradually interacts with the ocean under the action of gravity.

We conducted the first set of experiments on calm ocean. Figure 3 shows the simulation results. We simulated the movement of three oil spills with different viscosities under the same initial conditions (2.6×10^{-3} Pa s, 9.0×10^{-3} Pa s, and 1.0×10^{-1} Pa s). The density of oil (dark particles) is 840 kg/m³, and the density of ocean (blue particles) is 1027 kg/m³. It can be seen from the simulation results that the low viscosity oil has a larger spreading area and is easily broken into smaller oil films. The simulation results are basically consistent with Fay's assumption [6] that the oil film spreads in a circular shape on calm ocean. Compared with traditional methods, our DFPBF framework can simulate the natural breakup of the oil film.

```

for  $i$  in particles:
  search neighborhoods  $N_i(0)$ 
while  $t < t_{\max}$ :
  for  $i$  in particles:
    compute external forces:  $\mathbf{f}_{\text{ext}}$ 
  for  $i$  in particles:
    predict intermediate velocity  $\mathbf{v}_i^* = \mathbf{v}_i(t) + \Delta t \mathbf{f}_{\text{ext}}/m$ 
    predict intermediate position  $\mathbf{x}_i^* = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i^*$ 
  update time step size  $\Delta t$ 
  for  $i$  in particles:
    search neighborhoods  $N_i^*$ 
  for  $i$  in particles:
    compute reused factor  $\alpha_i^*$ 
   $\mathbf{x}^{\text{cd}} \leftarrow \text{ConstantDensityConstraint}(\alpha^*, \mathbf{x}^*)$ 
  for  $i$  in particles:
    update intermediate velocity  $\mathbf{v}_i^* = (1/\Delta t)(\mathbf{x}^{\text{cd}} - \mathbf{x}_i(t))$ 
   $\mathbf{v}^{\text{df}} \leftarrow \text{VelocityDivergenceConstraint}(\alpha, \mathbf{v}^*)$ 
   $\mathbf{v}_i(t + \Delta t) \leftarrow \text{ViscositySolver}(\mathbf{v}^{\text{df}})$ 
  for  $i$  in particles:
    update position  $\mathbf{x}_i(t + \Delta t) = \mathbf{x}^{\text{cd}}$ 

```

ALGORITHM 3: DFPBF simulation loop (calculation flow of the divergence-free position-based fluid (DFPBF) framework).

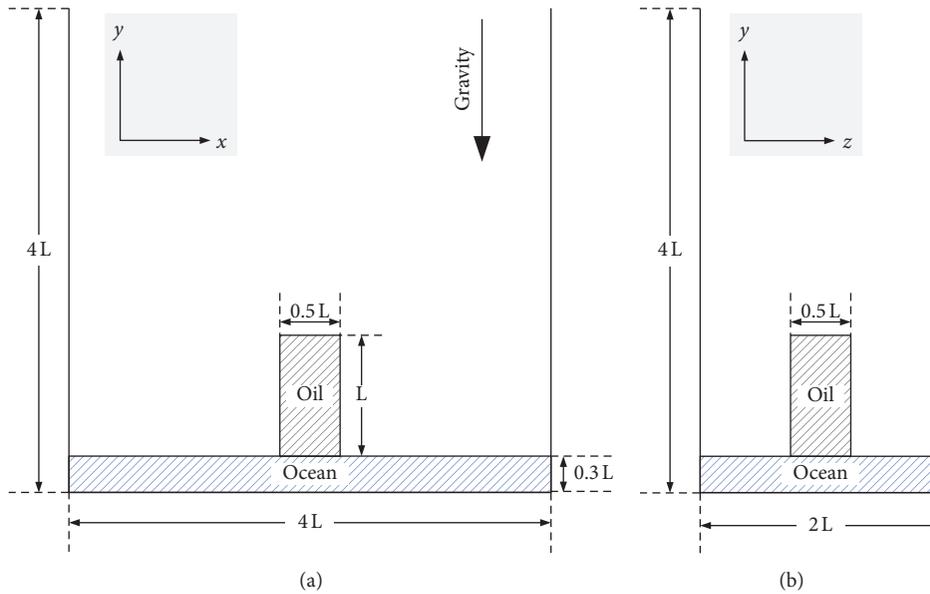


FIGURE 2: Schematic diagram of oil spill test scenario. The image on the left side is the front view of the scene, and the right image is the side view. All length units in this paper are dimensionless, and L denotes unit length. (a) Front view. (b) Side view.

We added a stochastic fluctuating wind field to the scene, as shown in Figure 3. The direction of the wind field is the negative direction along the x -axis. It can be seen from Figure 4 that the oil film drifts with the waves. Under the same wind wave conditions, the low viscosity oil spills are more affected by the wind waves. The shape of low-viscosity oil spreading on the ocean is more like an ellipse, with the long axis along the wind direction, which is consistent with Lehr's conclusion [8, 9]. In contrast, high-viscosity oil spills are more resistant to shear stress, so they are less affected by wind waves than low-viscosity oil spills.

Figure 5 is the coupling scene between oil spill and rock. The spilled oil flows out of the pipeline and the number of oil particles is 20k. The dynamic viscosity of oil in Figure 5 is 9.0×10^{-3} Pa s. Due to the viscosity, the spilled oil will stick to the rocks during the fall. When the oil spills drift to the nearshore zone, the animals, plants, and reefs may be coated by the dark crude oil with high viscosity. Our DFPBF framework with the implicit viscosity solver can realistically simulate these scenes, as shown in Figure 5.

We name the DFPBF framework which is using the relaxation scheme in Section 3.2 as rDFPBF (relaxed

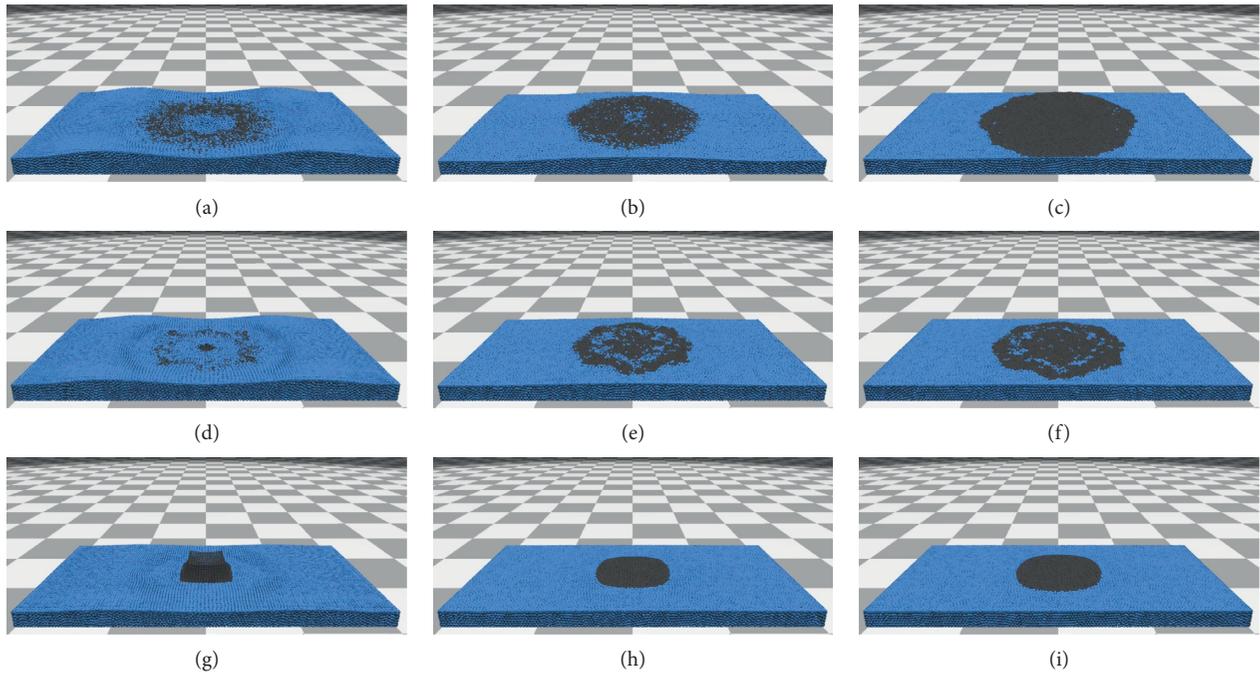


FIGURE 3: Simulation results of oil spill on calm ocean. The images in the first row are simulation results of oil with a dynamic viscosity of 2.6×10^{-3} Pa s at 1.0 s, 5.0 s, and 10.0 s, respectively. The dynamic viscosity of the oil in the second row is 9.0×10^{-3} Pa s and that in the third line is 1.0×10^{-1} Pa s. (a) $\mu = 2.6 \times 10^{-3}$ Pa s, $t = 1.0$ s, (b) $\mu = 2.6 \times 10^{-3}$ Pa s, $t = 5.0$ s, (c) $\mu = 2.6 \times 10^{-3}$ Pa s, $t = 10.0$ s, (d) $\mu = 9.0 \times 10^{-3}$ Pa s, $t = 1.0$ s, (e) $\mu = 9.0 \times 10^{-3}$ Pa s, $t = 5.0$ s, (f) $\mu = 9.0 \times 10^{-3}$ Pa s, $t = 10.0$ s, (g) $\mu = 1.0 \times 10^{-1}$ Pa s, $t = 1.0$ s, (h) $\mu = 1.0 \times 10^{-1}$ Pa s, $t = 5.0$ s, and (i) $\mu = 1.0 \times 10^{-1}$ Pa s, $t = 10.0$ s.

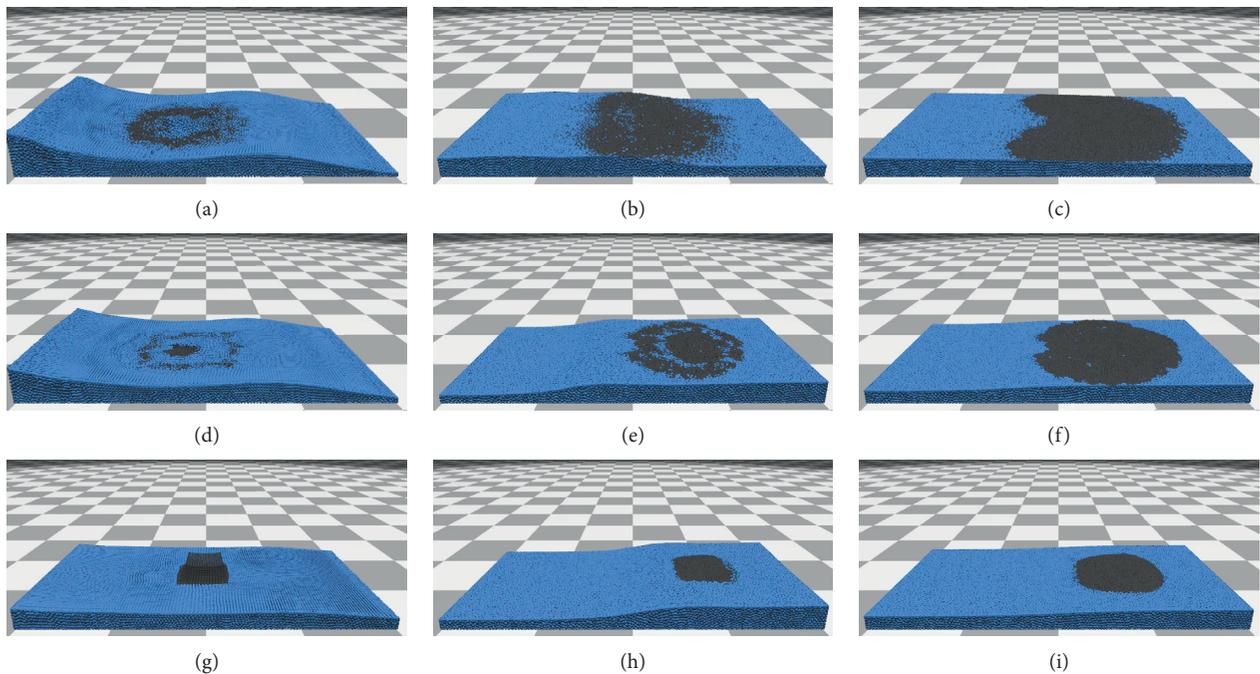


FIGURE 4: Simulation results of oil spill in wind and waves. The images in the first row are simulation results of oil with a dynamic viscosity of 2.6×10^{-3} Pa s at 1.0 s, 5.0 s, and 10.0 s, respectively. The dynamic viscosity of the oil in the second row is 9.0×10^{-3} Pa s and that in the third line is 1.0×10^{-1} Pa s. (a) $\mu = 2.6 \times 10^{-3}$ Pa s, $t = 1.0$ s, (b) $\mu = 2.6 \times 10^{-3}$ Pa s, $t = 5.0$ s, (c) $\mu = 2.6 \times 10^{-3}$ Pa s, $t = 10.0$ s, (d) $\mu = 9.0 \times 10^{-3}$ Pa s, $t = 1.0$ s, (e) $\mu = 9.0 \times 10^{-3}$ Pa s, $t = 5.0$ s, (f) $\mu = 9.0 \times 10^{-3}$ Pa s, $t = 10.0$ s, (g) $\mu = 1.0 \times 10^{-1}$ Pa s, $t = 1.0$ s, (h) $\mu = 1.0 \times 10^{-1}$ Pa s, $t = 5.0$ s, and (i) $\mu = 1.0 \times 10^{-1}$ Pa s, $t = 10.0$ s.

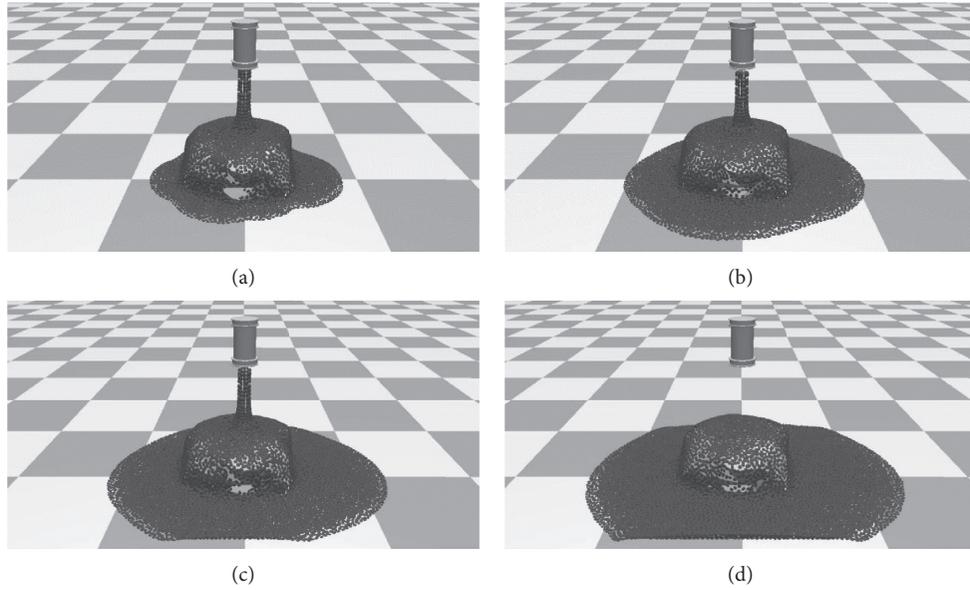


FIGURE 5: Coupling scene of oil spill and rock. Dark oil particles flow out of the pipeline. Images (a)–(d) are screenshots of oil spill scenes at 10.0 s, 15.0 s, 20.0 s, and 25.0 s, respectively. (a) $t = 10.0$ s. (b) $t = 15.0$ s. (c) $t = 20.0$ s. (d). $t = 25.0$ s.

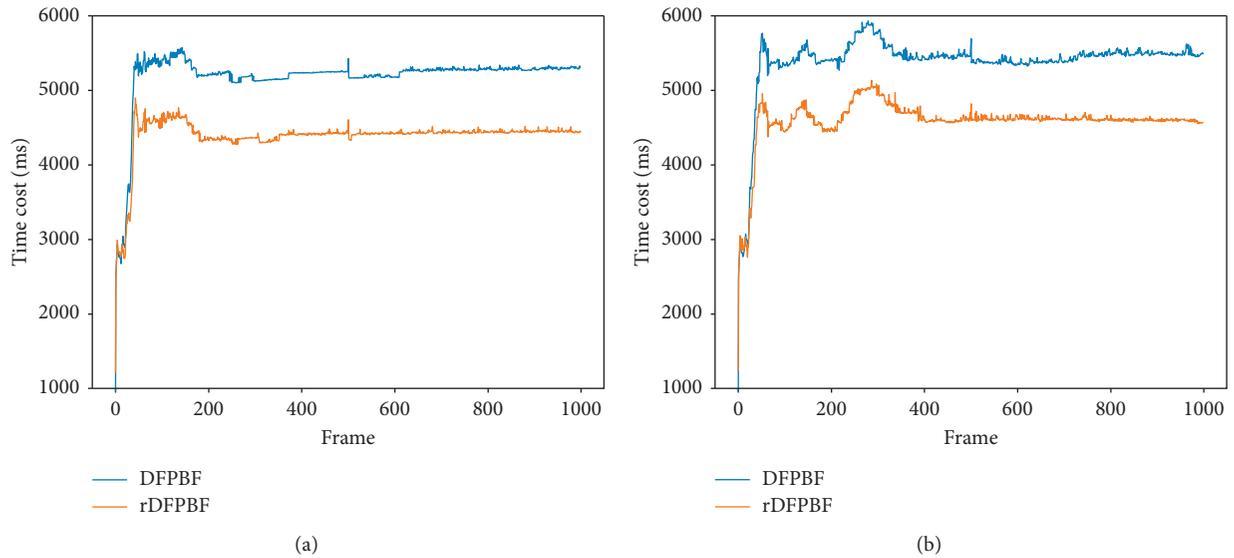


FIGURE 6: Time cost statistics for oil spill scenarios. Images (a) and (b) are, respectively, the time cost statistics of oil spill scenarios with different viscosities under the same initial conditions for 1000 frames (time steps). (a) $\mu = 2.6 \times 10^{-3} \text{ Pa} \cdot \text{s}$ (b) $\mu = 9.0 \times 10^{-3} \text{ Pa} \cdot \text{s}$.

DFPBF). Figure 6 shows the running time statistics of the oil spill scenarios shown in Figure 2. The time step size in this experiment is 0.01 s. It can be seen from Figure 6 that the rDFPBF framework shows attractive acceleration performance in both test scenarios. In Figure 6(a), the average time cost per time step of the DFPBF framework is 5.18 s and that of the rDFPBF framework is 4.38 s. In Figure 6(b), the average time cost per time step of DFPBF framework is 5.39 s and that of the rDFPBF framework is 4.59 s. As the viscosity of oil spills increases, the time cost also increases.

5. Conclusions

In this article, we proposed a meshless DFPBF oil spill simulation framework. The problem domain is discrete by Lagrangian particles, which is very suitable for simulating the breaking and merging motion of the oil film. The particle number density model was used to calculate the physical quantities of Lagrangian particles, which can accurately estimate the physical quantities and handle density discontinuities at the interface between multiple fluids with varying rest

densities. We also introduced an implicit viscosity solver for our DFPBF framework, which can stably simulate oil spills with various viscosities. We conducted a series of simulation experiments based on our DFPBF framework, and the following conclusions can be drawn based on the experimental results:

- (1) On a calm ocean, the oil spills will spread approximately into a circular shape, and as the viscosity increases, the spreading area gradually decreases. In this respect, our conclusion is consistent with Fay's conclusion [5, 6].
- (2) Under wind wave conditions, the oil spill will drift with the wind waves, and the final position will deviate from the initial oil spill position. The spreading shape of the oil spill is more like an ellipse, and the long axis of the ellipse is in the direction of the wind waves, which is consistent with Lehr's conclusion [8, 9].
- (3) Our oil spill model can simulate the scene of oil spills covering reefs, which helps improve the trainees' emergency response capability to deal with such oil spills.

We have made efforts to improve the performance of the DFPBF framework, such as adaptive time step size, matrix-free conjugate gradient method, and Jacobi preconditioner. When applied to marine simulators, parallel computing and distributed computing can be used to further improve the performance. Since our DFPBF framework is only designed for virtual scenes in marine simulators and only used to train trainees' emergency response ability, we are more inclined to improve its performance than the accuracy of the algorithm. Therefore, our DFPBF framework cannot be directly introduced to oil spill forecasting. If researchers try to extend our method in oil spill forecasting, it is necessary to compare the simulated results with the measured data. Moreover, some parts of our DFPBF framework need to adopt higher-precision scheme, such as changing Euler integration to higher-order Runge–Kutta and changing the kernel function from cubic spline to a more precise form.

In future work, we will try to introduce surface tension [45] and turbulence effects [46] to our DFPBF framework in order to simulate oil spill spreading and drift transport with Langmuir circulation. In terms of interactive simulation of boom and oil spill, we try to model the boom based on the PBD method to further enhance the realism.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors thank the financial support provided by National Natural Science Foundation of China (Grant no.

52071312), Liaoning Provincial Natural Science Foundation shipping joint fund project (Grant no. 2020-HYLH-29), and Provincial postgraduate education and teaching achievement award cultivation project (Grant no. YJG2020601).

References

- [1] K. Li, H. Yu, J. Yan, and J. Liao, "Analysis of offshore oil spill pollution treatment technology," *IOP Conference Series: Earth and Environmental Science*, vol. 510, Article ID 042011, 2020.
- [2] V. Do and H. Ren, "Using Poisson disk sampling to render oil particles at sea," *Journal of Computing and Information Technology*, vol. 27, no. 4, pp. 31–46, 2019.
- [3] H. Li, *Physical model-based ocean scene modeling in marine simulator*, PhD Thesis, Dalian Maritime University, Dalian, China, 2020.
- [4] ITOPF, "Fate of marine oil spills," 2020, <https://www.itopf.org/knowledge-resources/documents-guides/document/tip-02-fate-of-marine-oil-spills/>.
- [5] J. A. Fay, "The spread of oil slicks on A calm sea," in *Oil on the Sea*, D. P. Hoult, Ed., vol. 6, pp. 53–63, Springer, Boston, MA, USA, 1969.
- [6] J. A. Fay, "Physical processes in the spread of oil on A water surface," in *Proceedings of the 2005 International Oil Spill Conference*, pp. 463–467, Global Engineering Documents, Miami Beach, FL, USA, May 2005.
- [7] D. Mackay, K. Trudel, and S. Paterson, *A Mathematical Model of Oil Spill Behaviour*, pp. 1–12, Research and Development Division, Environmental Emergency Branch, Ottawa, Canada, 1980.
- [8] W. Lehr and D. Simecek-Beatty, "The relation of Langmuir circulation processes to the standard oil spill spreading, dispersion, and transport algorithms," *Spill Science & Technology Bulletin*, vol. 6, no. 3-4, pp. 247–253, 2000.
- [9] W. J. Lehr, "Review of modeling procedures for oil spill weathering behavior," *Advances in Ecological Research*, vol. 9, pp. 51–90, 2001.
- [10] D. Simecek-Beatty and W. J. Lehr, "Extended oil spill spreading with Langmuir circulation," *Marine Pollution Bulletin*, vol. 122, no. 1-2, pp. 226–235, 2017.
- [11] F. Yu, *Study on the Real-Time Simulation and Visualization Offshore Oil Spill in Marine Search and Rescue Simulator*, Dalian Maritime University, Dalian, China, 2010.
- [12] A. Ribotti, F. Antognarelli, A. Cucco et al., "An operational marine oil spill forecasting tool for the management of emergencies in the Italian seas," *Journal of Marine Science and Engineering*, vol. 7, p. 1, 2019.
- [13] G. Zanier, M. Palma, A. Petronio, F. Roman, and V. Armenio, "Oil spill scenarios in the kotor bay: results from high resolution numerical simulations," *Journal of Marine Science and Engineering*, vol. 7, no. 2, p. 54, 2019.
- [14] M. De Dominicis, N. Pinardi, G. Zodiatis, and R. Lardner, "MEDSLIK-II, a Lagrangian marine surface oil spill model for short-term forecasting - Part 1: Theory," *Geoscientific Model Development*, vol. 6, no. 6, pp. 1851–1869, 2013.
- [15] M. De Dominicis, N. Pinardi, G. Zodiatis, and R. Archetti, "MEDSLIK-II, a Lagrangian marine surface oil spill model for short-term forecasting - Part 2: numerical simulations and validations," *Geoscientific Model Development*, vol. 6, no. 6, pp. 1871–1888, 2013.
- [16] P. Tkalic, "A CFD solution of oil spill problems," *Environmental Modelling & Software*, vol. 21, no. 2, pp. 271–282, 2006.

- [17] A. Amini and A. J. Schleiss, "Numerical modeling of oil-water multiphase flow contained by an oil spill barrier," *Engineering Applications of Computational Fluid Mechanics*, vol. 3, no. 2, pp. 207–219, 2009.
- [18] D. M. Li, Z. Zhen, H. Q. Zhang, Y. Q. Li, and X. C. Tang, "Numerical model of oil film diffusion in water based on SPH method," *Mathematical Problems in Engineering*, vol. 2019, Article ID 8250539, 14 pages, 2019.
- [19] D. Violeau, C. Buvat, K. Abed-Meraim, and E. De Nanteuil, "Numerical modelling of boom and oil spill with SPH," *Coastal Engineering*, vol. 54, no. 12, pp. 895–913, 2007.
- [20] X. Yang and M. Liu, "Numerical modeling of oil spill containment by boom using SPH," *Science China Physics, Mechanics and Astronomy*, vol. 56, no. 2, pp. 315–321, 2013.
- [21] G. Duan, B. Chen, X. Zhang, and Y. Wang, "A multiphase MPS solver for modeling multi-fluid interaction with free surface and its application in oil spill," *Computer Methods in Applied Mechanics and Engineering*, vol. 320, pp. 133–161, 2017.
- [22] M. Janati, M. Kolahdoozan, and H. Imanian, "Artificial neural network modeling for the management of oil slick transport in the marine environments," *Pollution*, vol. 6, pp. 399–415, 2020.
- [23] F. Ferstl, R. Ando, C. Wojtan, R. Westermann, and N. Thuerey, "Narrow band FLIP for liquid simulations," *Computer Graphics Forum*, vol. 35, no. 2, pp. 225–232, 2016.
- [24] C. J. Zou, *Emergency Simulation and Visualization of Oil Spill from Ship in Navigation Simulation*, Dalian Maritime University, Dalian, China, 2018.
- [25] D. P. Hoult, "Oil spreading on the sea," *Annual Review of Fluid Mechanics*, vol. 4, no. 1, pp. 341–368, 1972.
- [26] G. Coppini, M. D. Dominicis, G. Zodiatis et al., "Hindcast of oil-spill pollution during the Lebanon crisis in the eastern mediterranean," *Marine Pollution Bulletin*, vol. 62, pp. 140–153, 2011.
- [27] R. W. Lardner and G. Zodiatis, "An operational oil spill model in the levantine basin (eastern mediterranean sea)," *Marine Pollution Bulletin*, vol. 10, pp. 5–9, 1988.
- [28] D. De Padova, M. Mossa, M. Adamo, G. De Carolis, and G. Pasquariello, "Synergistic use of an oil drift model and remote sensing observations for oil spill monitoring," *Environmental Science and Pollution Research*, vol. 24, no. 6, pp. 5530–5543, 2017.
- [29] K.-F. Dagestad, J. Röhrs, Ø. Breivik, and B. Ådlandsvik, "OpenDrift v1.0: a generic framework for trajectory modeling," *Geoscientific Model Development*, vol. 11, no. 4, pp. 1405–1420, 2018.
- [30] J. Röhrs, K.-F. Dagestad, H. Asbjørnsen et al., "The effect of vertical mixing on the horizontal drift of oil spills," *Ocean Science*, vol. 14, no. 6, pp. 1581–1601, 2018.
- [31] P. Keramea, K. Spanoudaki, G. Zodiatis, G. Gikas, and G. Sylaios, "Oil spill modeling: a critical review on current trends, perspectives, and challenges," *Journal of Marine Science and Engineering*, vol. 9, no. 2, p. 181, 2021.
- [32] D. J. Price, "Smoothed particle hydrodynamics and magnetohydrodynamics," *Journal of Computational Physics*, vol. 231, no. 3, pp. 759–794, 2012.
- [33] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, CA, USA, July 2003.
- [34] L. Brookshaw, "A method of calculating radiative heat diffusion in particle simulations," *Publications of the Astronomical Society of Australia*, vol. 6, no. 2, pp. 207–210, 1985.
- [35] J. J. Monaghan, "On the problem of penetration in particle methods," *Journal of Computational Physics*, vol. 82, no. 1, pp. 1–15, 1989.
- [36] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annual Review of Astronomy and Astrophysics*, vol. 30, no. 1, pp. 543–574, 1992.
- [37] H. Schechter and R. Bridson, "Ghost SPH for animating water," *ACM Transactions on Graphics*, vol. 31, p. 61, 2012.
- [38] M. Macklin and M. Müller, "Position based fluids," *ACM Transactions on Graphics*, vol. 32, p. 104, 2013.
- [39] T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, and M. C. Lin, "Implicit formulation for SPH-based viscous fluids," *Computer Graphics Forum*, vol. 34, no. 2, pp. 493–502, 2015.
- [40] A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner, "An implicit viscosity formulation for SPH fluids," *ACM Transactions on Graphics*, vol. 34, p. 114, 2015.
- [41] J. Bender and D. Koschier, "Divergence-free SPH for incompressible and viscous fluids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 1193–1206, 2016.
- [42] M. Weiler, D. Koschier, M. Brand, and J. Bender, "A physically consistent implicit viscosity solver for SPH fluids," *Computer Graphics Forum*, vol. 37, no. 2, pp. 145–155, 2018.
- [43] B. Solenthaler and R. Pajarola, "Density contrast SPH interfaces," in *Proceedings of the 8th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Association for Computing Machinery, Dublin, Ireland, July 2008.
- [44] X. Wang, Y. Xu, X. Ban, S. Liu, and Y. Xu, "A unified multiple-phase fluids framework using asymmetric surface extraction and the modified density model," *Symmetry*, vol. 11, no. 6, p. 745, 2019.
- [45] X. He, H. Wang, F. Zhang, H. Wang, G. Wang, and K. Zhou, "Robust simulation of sparsely sampled thin features in SPH-based free surface flows," *ACM Transactions on Graphics*, vol. 34, p. 7, 2014.
- [46] X. Wang, S. Liu, X. Ban, Y. Xu, J. Zhou, and J. Kosinka, "Robust turbulence simulation for particle-based fluids using the rankine vortex model," *The Visual Computer*, vol. 36, pp. 2285–2298, 2020.