

Article

Real-Time Smoke Simulation based on Vortex Method

Fei Wan ^{1,*†}, Jingpu Zhang ^{1,†}, Lizheng Guo ^{1,†} and Yunchang Liu ^{1,†}¹ School of Computer and Data Science, Henan University of Urban Construction, Pingdingshan, Henan, China; 85997558@qq.com; 20181027@hncj.edu.cn; kftjhpds@163.com; yunchang75@aliyun.com

* Correspondence: 85997558@qq.com; Tel.: +86-13592173767

† These authors contributed equally to this work.

Abstract: In this paper, we use three different experimental methods (particle method, grid method and hybrid method) to model and simulate the smoke from the perspective of fluid dynamics. Through the comparison of different methods, we conclude: The particle method can avoid the numerical dissipation problem caused by grid calculation, but it also brings problems such as the distortion of the trajectory of the example. The grid method is accurate in calculation, but it is prone to numerical dissipation and loss of details. Finally, we choose the hybrid method to store the vorticity in the form of particles in vortex particles, avoiding the numerical dissipation problem caused by the use of grids, and including rich turbulence, which perfectly shows the simulation effect of smoke.

Keywords: smoke simulation; vortex method; Vortex-In-Cell (VIC)

1. Introduction

Physics based smoke animation has been simulated for many years. In the early days, researchers used the Navier-Stokes equation to speed the motion in the graph. For example, Foster and Metaxes[1] introduced fluid simulation using the finite difference method into the graph for the first time, Stam[2] introduced a semi-Lagrangian advection scheme to achieve unconditional stability of the fluid, Fedkiw et al.[3] applied fluid simulation to similar smoke and added vortex control to enhance vortex detail. Based on this, Kim et al.[4] and Selle et al.[5] gave a higher order interpolation algorithm to improve its accuracy later. Zhu and Bridson[6] introduced FLIP to concentrate the advection term on the particles and discard the previous numerical dissipation.

Compared with other fluid animation, the simulation of smoke animation pays more attention to the display of surface vorticity. Later, some scholars introduced vortex particle methods, which can avoid the need to solve the elliptic equation separately in the velocity-pressure equation. The vorticity-based fluid simulation mainly includes the brute force direct summation of the N-body problem of the particle, and a series of treecode[7], FMM[8], etc. These methods all show wonderful effects, but also bring problems such as flow field distortion. At the same time, at each time step, the new position of the N vortices is calculated, and $O(N^2)$ operations are required. Due to limitations of computer resources and costs, it is impossible to make N too large. In order to improve the effect of vortex methods to simulate the fluid, a larger number of vortices are needed, and when the number of vortices is large, the chaotic state is prone to occur.

To overcome these shortcomings, we choose the Vortex-In-Cell method (VIC), which was first proposed by Christiansen [9]. We know that the vortex particle method is a no-grid Lagrange method. The VIC method is a hybrid method of semi-Lagrange and semi-Euler. In the numerical simulation of the vorticity field, the particle tracking Lagrange method is still maintained, but in the processing of the velocity field, we adopt the Euler method with grid. First, the velocity field is calculated by the Poisson equation of the flow function in the Euler frame, and then the vortex element is tracked along the trace in the Lagrange frame. This idea can be traced back to the MAC method of Harlow and Welch [10].



Citation: Wan, F.; Zhang, J.; Guo, L.; Liu Y. Title. *Preprints* **2021**, *1*, 0.
<https://doi.org/>

Received:

Accepted:

Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

2. Related Work

In this paper, we introduce three methods, including particle method, grid method and hybrid method:

Particle method: Selle et al. [11] used vortex particles to augment a more traditional Eulerian velocity-pressure simulation. Pfaff et al. [12] used vortex particles from pre-processed boundary layers to synthesize turbulence. Weißmann and Pinkall [13] used Biot-Savart summation for vortex rings. Brochu et al. [14] captured vorticity with a Lagrangian mesh, including its generation from buoyancy, and used FMM for lineartime Biot-Savart summation. Lagrangian vorticity-based fluid solvers [15] evaluate the velocity field by summing over vortex sources with the Biot-Savart law. Berlin noise [16] etc.

Grid method: In fluid simulation, the common advection is the advection of velocity, such as Foster and Metaxas [1] used the Euler method to simulate the advection of velocity, and Stam [2] introduced the semi-Lagrangian method, by using linear backtracking, make it unconditionally stable. Fedkiw et al. [3] introduced the semi-Lagrangian method into the smoke simulation for solving the advection of temperature and density. Based on this, Kim et al., [4], Selle et al. [5] gave a higher order interpolation algorithm to improve its accuracy by using BFECC scheme and MacComark method respectively. With regard to the semi-Lagrangian method for vorticity advection, mainly for solving atmospheric models [17], contour [18,19], etc.

Hybrid method: Zhang and Bridson [20] used vortex segments to simulate vortex stretching. Marshall and Grant [21], Liu and Doorly [22] used the VIC method to directly calculate vortex stretching on a temporary Euler grid. Selle et al. [11] determined the stretching term by given velocity fields and grids. Zhang et al. [23] simplified the item and no longer used complex Jacobian matrices. Graham [24] first used the explicit Euler method to solve the diffusion term in the viscous solution 2D VIC method. Cheng et al. [25] used the "diffusion-vortex" scheme proposed by Lu and Ross [26] to solve the flow on a circular cylinder in a 2D VIC solver.

3. Theory

3.1. Hydrodynamic equations

3.1.1. Dynamic equation of temperature and density

In a system, if there are no other external factors, assuming that the temperature and density changes of the fluid element are represented by the material derivative DT/Dt and $D\rho/Dt$ respectively, we can obtain the simplest equation

Lagrangian form:

$$\begin{aligned}\frac{DT}{Dt} &= k_T \nabla^2 T \\ \frac{D\rho}{Dt} &= k_\rho \nabla^2 \rho\end{aligned}\quad (1)$$

Euler form:

$$\begin{aligned}\frac{\partial T}{\partial t} + u \cdot \nabla T &= k_T \nabla^2 T \\ \frac{\partial \rho}{\partial t} + u \cdot \nabla \rho &= k_\rho \nabla^2 \rho\end{aligned}\quad (2)$$

where k_T and k_ρ are non-negative diffusion coefficients.

3.1.2. Vortex equation

According to Boussinesq approximation [27]: A small change in density has no effect on the inertia and viscosity terms in the fluid motion equation, and the only effect is reflected in the buoyancy generated. Therefore, according to this kind of movement mechanism, combined with Navier-Stocks equation [1], we construct the fluid movement model as follows:

Euler form:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = \nu \nabla^2 u - \nabla p + F_{bv} \quad (3)$$

Lagrangian form:

$$\frac{Du}{Dt} = \nu \nabla^2 u - \nabla p + F_{bv} \quad (4)$$

where u is the ascending velocity of the fluid, ν is the viscosity coefficient, p is the pressure, and F_{bv} is the buoyancy of the fluid.

Then we find the curl ($\nabla \times$) on both sides of the Equation (3), can get the vortex equation about vorticity ω

Euler form:

$$\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega = \omega \cdot \nabla u + \nu \nabla^2 \omega + \nabla \times F_{bv} \quad (5)$$

Lagrangian form:

$$\frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega + \nabla \times F_{bv} \quad (6)$$

3.1.3. External forces

Compared with the surrounding air, the smoke fluid has the characteristics of high temperature and low density. Assuming that the unit volume is 1, the buoyancy of the smoke is:

$$F_{bv} = -g \frac{\rho_0}{\rho} \quad (7)$$

Where ρ_0 is the density of the air, ρ is the density of the smoke fluid, and g is the acceleration of gravity.

At the same time, according to the thermal expansion performance of the fluid, the buoyancy of the smoke rising process can also be expressed as

$$F_{bv} = -\beta g (T_0 - T) \quad (8)$$

where temperature of smoke is T , the temperature of air is T_0 , g is the acceleration of gravity and β is the coefficient of thermal expansion.

Therefore, the external force exerted by the smoke rising process is

$$F_{bv} = -g \frac{\rho_0}{\rho} - \beta g (T_0 - T) \quad (9)$$

The vorticity is mainly generated by viscous forces and baroclinic fluids. Viscous force is mainly the interaction between fluid and solid, while baroclinic fluid forces is mainly caused by uneven density distribution between different fluids, as shown in Figure 1(a):

In fluid A (smoke), the density is ρ , the pressure is p , the flow velocity is u upward due to buoyancy; In fluid B (air), the density is ρ_0 , the pressure is p_0 . Fluid A has a higher density and pressure due to its higher heating temperature, and the density $\rho < \rho_0$, pressure $p < p_0$. So the lower fluid B flows to the left while the upper portion starts to move to the right due to the increase of the density of fluid A , thereby causing the generation of vorticity [28]. At the same time, during the ascent, all vortices at the boundary of the smoke fluid and the surrounding fluid form a vortex ring (shown by the dotted blue line in Figure 1(b)). Meanwhile, Figure 1(b) shows the vorticity distribution produced by rising smoke. The value of vorticity is the curl of buoyancy of smoke rising ($\nabla \times F_{bv}$).

3.2. Application of semi-Lagrangian method in fluid field

The implicit semi-Lagrangian method can make the step size larger or the grid spacing smaller, which can achieve unconditional stability.

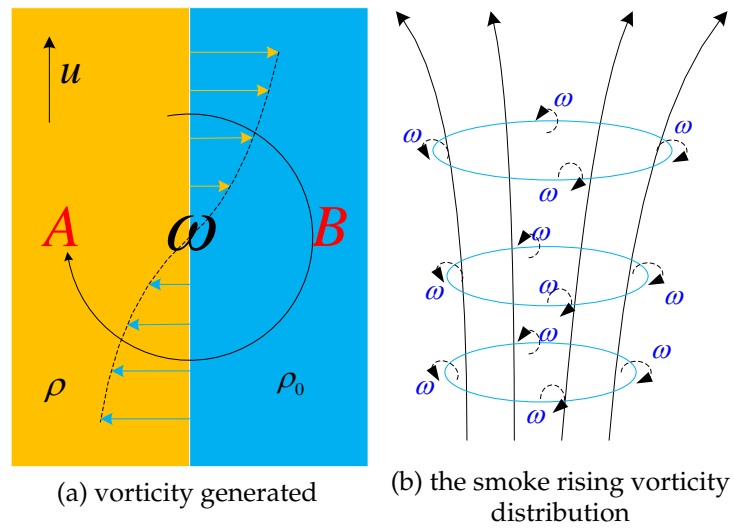


Figure 1. The vorticity generated by rising smoke

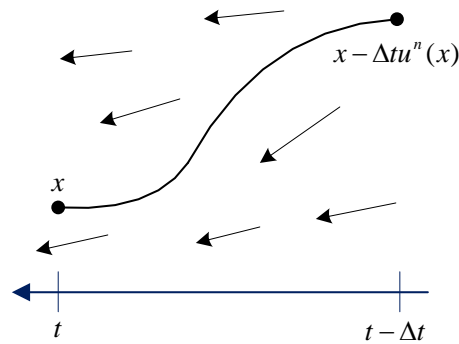


Figure 2. Linear backtracking

3.2.1. Advection term

Take the advection term of the temperature field as an example,

$$u \cdot \nabla T = \frac{\partial T}{\partial u} = \lim_{h \rightarrow 0} \frac{T(x + hu) - T(x)}{h} \quad (10)$$

Then

$$-u \cdot \nabla T \approx \frac{T(x - \Delta t u^n(x)) - T(x)}{\Delta t} \quad (11)$$

According to Equation (2),

$$T^{n+1}(x) = T^n(x) + \Delta t \frac{T^n(x - \Delta t u^n(x)) - T^n(x)}{\Delta t} \quad (12)$$

$$T^{n+1}(x) = T^n(x - \Delta t u^n(x)) \quad (13)$$

Similarly, to derive the density field and vorticity field by using the linear backtracking method, we can get

$$\rho^{n+1}(x) = \rho^n(x - \Delta t u^n(x)) \quad (14)$$

$$\omega^{n+1}(x) = \omega^n(x - \Delta t u^n(x)) \quad (15)$$

Figure 2 shows a process in which the vorticity at time t and at position x can be linearly traced back to the vorticity at time $t - \Delta t$ and at position $x - \Delta t u^n(x)$.

According to Equation (13), as long as the linear interpolation algorithm is used, there must be $|T_{max}^{n+1}| = |T_{max}^n|$. The advection method of this item is unconditionally stable, that

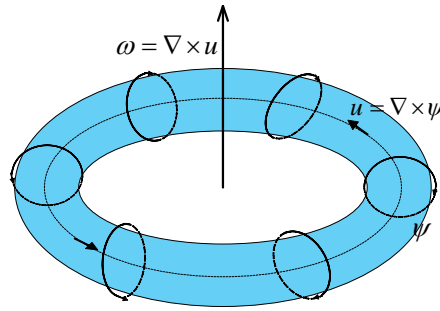


Figure 3. The relationship between vorticity ω , velocity u and stream function ψ

is, no matter how large Δt is, or how fast u is, the final interpolation range must be within the range of the maximum value of the previous step. Therefore, the vorticity field will only decrease without increasing the external force.

3.2.2. Diffusion term

Meanwhile, in the diffusion term of the temperature field, for larger viscosity or smaller grid size, the implicit semi-Lagrangian method is also used, and the following form is changed:

$$T^{n+1}(x) - k_T \Delta t \nabla^2 T^{n+1}(x) = T^n(x) \quad (16)$$

Then

$$(I - k_T \Delta t \nabla^2) T^{n+1}(x) = T^n(x) \quad (17)$$

Where I is the identity matrix.

Similarly, in the density field and vorticity field,

$$(I - k_\rho \Delta t \nabla^2) \rho^{n+1}(x) = \rho^n(x) \quad (18)$$

$$(I - \nu \Delta t \nabla^2) \omega^{n+1}(x) = \omega^n(x) \quad (19)$$

3.3. Smoke modeling based on vorticity field

3.3.1. Induced velocity field of vorticity field

For the incompressible vector velocity field u , we define the stream function vector ψ , and $u = \nabla \times \psi$. The relationship between vorticity ω , velocity u and stream function ψ is shown in Figure 3. Then

$$\omega = \nabla \times u = \nabla \times \nabla \times \psi = \nabla(\nabla \cdot \psi) - \nabla^2 \psi \quad (20)$$

By the limitation of $\nabla \cdot \omega = 0$, it can be assumed that the divergence of the stream function in the vorticity field is also 0, that is $\nabla \cdot \psi = 0$, then

$$\nabla^2 \psi = -\omega \quad (21)$$

The equation is a Poisson equation, which can be solved iteratively by the finite difference method. After ψ is obtained, the value of u can be obtained from equation $u = \nabla \times \psi$.

When we use the particle method, in 3D space, according to the Green function method, we can get the stream function vector

$$\psi(r) = \frac{1}{4\pi} \int \frac{\omega}{r} dV \quad (22)$$

Where r is the distance and dV is the volume microelement, so u can be expressed as

$$u = \nabla \times \psi = -\frac{1}{4\pi} \int \omega \times \nabla \frac{1}{r} dV \quad (23)$$

Which is

$$u = \frac{1}{4\pi} \int \frac{\omega \times r}{r^3} dV \quad (24)$$

Then discretize it to get the velocity solution, namely

$$u_i = \sum_{j,j \neq i}^N \frac{\omega_j \times (r_i - r_j)}{4\pi r_{ij}^3} \quad (25)$$

Where r_{ij} is the distance between the i th vortex particle and the j th vortex particle.

Equation (24) and Equation (25) are the analytical and discrete forms of the Biot-Savart law, respectively. It can be seen from the Equation (25) that the velocity of each particle is obtained by summing the vorticity of all other particles in the space. This problem has the same form as solving the universal gravitation in the universe, which is also the N-Body problem.

3.3.2. Vortex-In-Cell (VIC) method

The Vortex-In-Cell (VIC) method is to use the Poisson Equation (21) of the Euler coordinate system flow function to calculate the velocity field, and then track the vortex particles along the trace in the Lagrange coordinate system.

First, we divide the grid on the computing plane. In order to solve the Poisson equation, we distribute the vorticity carried by vortices to the corresponding grid nodes. When we use the area weighting method, as shown in Figure 4.

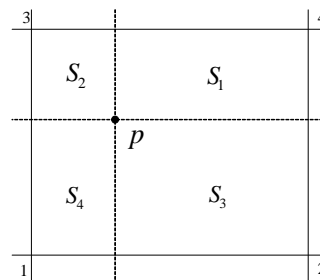


Figure 4. Area weighting method

From Figure 4, it can be seen that the vorticity ω_p of the vortex particles located in the grid is distributed to the surrounding grid points by means of area weighting (Figure 4 in 1, 2, 3, 4). The proportion of each node can be allocated according to the size of its diagonal area, respectively, S_1, S_2, S_3, S_4 , extended to three dimensions, the vorticity on each node

$$\omega(k) = \omega_p S(k) \quad k = 1, 2, 3, 4, 5, 6, 7, 8 \quad (26)$$

After the vorticity of all vortex particles is distributed to each node of the corresponding grid according to the above method, if n particles of one grid node are adjacent to each other, then the vorticity of each grid node ω is

$$\omega = \sum_{k=1}^n \omega(k) \quad (27)$$

At the same time, the sum of the area weight W at this node is

$$W = \sum_{k=1}^n S(k) \quad (28)$$

After obtaining the vorticity ω on the grid nodes, Equation (21) can be solved by the finite difference method (FDM), including Jacobian iteration, Gauss-Seidel iteration, conjugate gradient method, and multigrid method, etc. After obtaining ψ , the value of u can be obtained according to $u = \nabla \times \psi$.

The next step is to track the motion of vortex particles in the Lagrangian framework. According to the velocity on the grid node, the velocity of each vortex particle can be obtained by area weighted interpolation

$$u_p = \sum_{k=0}^8 \frac{u(k)}{W(k)} S(k) \quad (29)$$

From this, the position of each vortex particle at the next moment can be calculated.

The biggest advantage of the Vortex-In-Cell method is the high computational efficiency of the velocity field. Each time step only needs to calculate $O(N) + O(M \log_2 M)$ times, where M is the number of grids, which is one order of magnitude smaller than $O(N^2)$ in the direct summation method. In this way, we can introduce a much larger number of vortex particles and improve the resolution of the flow field, especially in the area where the vorticity is concentrated, and can give full play to the advantages of Lagrangian and Euler methods. Because the grid vortex method is a method that combines grids and particles, compared with the pure finite difference grid method, this method is not constrained by the stability determination conditions when dealing with vorticity convection. That is to say, no matter what the time step is, it can be unconditionally stable.

3.3.3. Vorticity stretching

In 3D fluids, the interaction between the local vorticity vector and the velocity gradient results in vorticity stretching.

(1) If adopt the particle method, we can use the vortex segment method instead [20]. The vortex segment can be regarded as a cylinder rotating around the central axis. The direction of the central axis is consistent with the direction of the vortex. It has two endpoints x_a and x_b , length h , and curl κ . A vortex particle with a vortex intensity of ω_i can have a length of h_i and a unit direction of $\hat{t} = \omega_i / \sqrt{\|\omega_i\|^2}$, $\text{curl } \kappa = \omega / (h_i \hat{t}_i)$ represents a vortex segment. At the beginning of each time step, the vortex particles are converted into such vortex segments, and then the speed at both ends of the vortex segment is calculated, and the positions of both ends are moved.

$$\begin{aligned} x_{i,a}^{n+1} &= x_{i,a}^n + \Delta t u_{i,a}^n \\ x_{i,b}^{n+1} &= x_{i,b}^n + \Delta t u_{i,b}^n \end{aligned} \quad (30)$$

When both ends of the vortex segment are updated, the vortex segment is stretched and converted back to vortex particles with vorticity

$$\omega_i = \kappa (x_{i,b}^{n+1} - x_{i,a}^{n+1}) \quad (31)$$

It can be found that the process is cyclic. The whole process can be represented by Figure 5 as

(2) If we adopt grid method, when the vortex element passes through the advection, it is simultaneously stretched by the velocity field, changing the vorticity field. Through the separation of the Euler equation, we need to solve

$$\omega \cdot \nabla u = \frac{\partial u}{\partial \omega} = \lim_{\varepsilon \rightarrow 0} \frac{u(x + \varepsilon \omega / 2) - u(x - \varepsilon \omega / 2)}{\varepsilon} \quad (32)$$

From this, we can obtain a new vorticity

$$\omega^{n+1}(x) = \omega^n(x) + \frac{\Delta t}{\varepsilon} [u^n(x + \varepsilon \omega^n(x) / 2) - u^n(x - \varepsilon \omega^n(x) / 2)] \quad (33)$$

In the calculation, if the grid scale is selected as h , then $\varepsilon = \frac{h}{\|\omega\|_2}$. In this way, the velocity can be sampled at both ends of the grid to better show how the vorticity field is stretched by the velocity field.

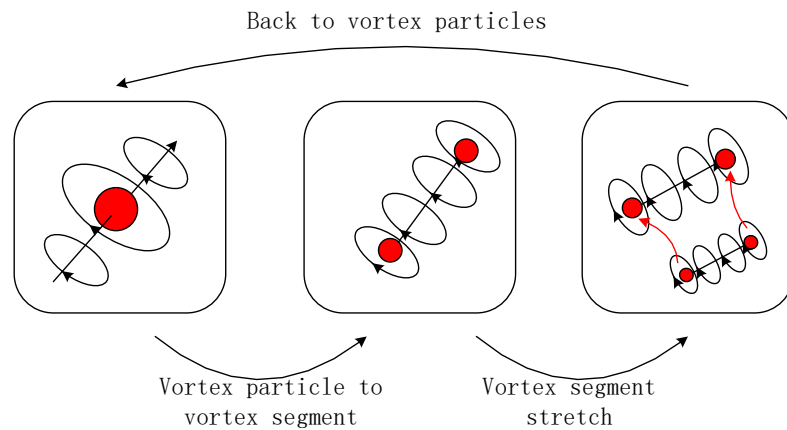


Figure 5. Transition from vortex blobs to vortex segments

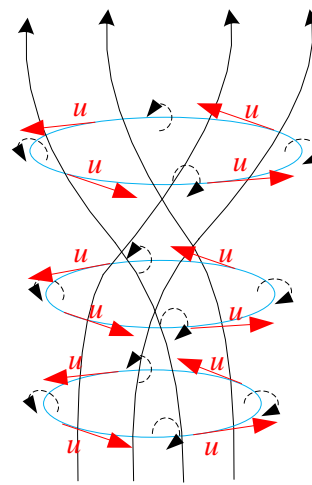


Figure 6. Vorticity stretching

It can be seen from Equation (33) that the velocity increment generated by the stretching term at a point in the vortex ring coincides with the direction of the vorticity of the point. As shown in Figure 6, the red arrow in the figure indicates the velocity increment caused by the stretching, which coincides with the direction of the vorticity at that point. At the same time, due to the stretching of the vorticity, the trajectory of the smoke flow is also bent (shown by the solid black line in Figure 6).

4. Implementation process

In this article, we have established three smoke models, using the particle method based on N-body summation, the semi-Lagrangian vortex method based on the grid, and the Vortex-In-Cell method based on the hybrid of particles and grids. The following are the flowcharts, algorithms and detailed steps of the three methods. In the three flowcharts, the round box \bigcirc represents the calculation based on particles, and the rectangular box \square represents grid-based computing. In order to satisfy the above vector field at the same time, we have select particles that contain temperature T_p , density ρ_p , vorticity ω_p and tracking particles with velocity u_p . At the same time, the variables on the grid include: temperature T , density ρ , vorticity ω , velocity u and flow function vector ψ .

4.1. Particle method (N-body method)

In the particle method, the model uses the N-body summation method based on the Biot-Sarvat law, which all calculations of temperature, density, and vorticity are based on particles. This method can avoid the numerical dissipation problem caused by grid

calculation, but at the same time it brings about the distortion of particle trajectory and other problems. The flowchart is shown in Figure 7. The implementation process is shown

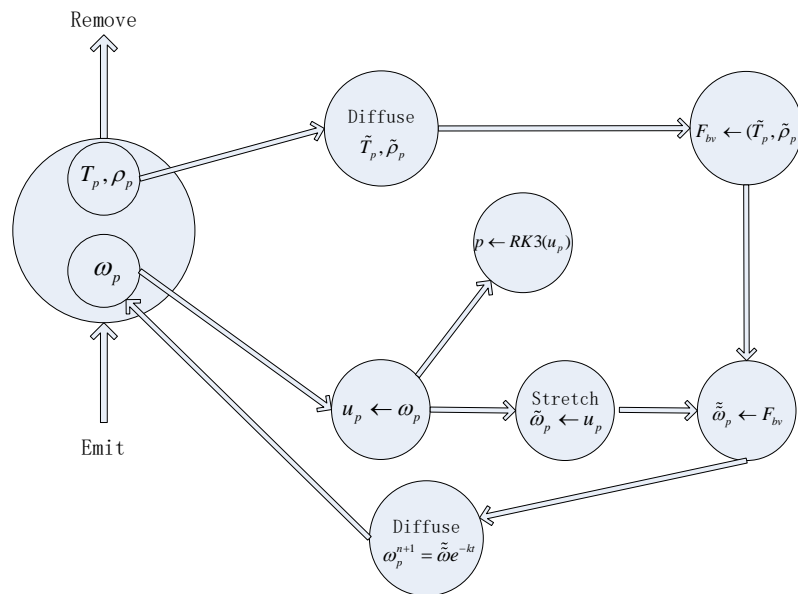


Figure 7. Flow chart of smoke (particle method)

in Algorithm 1:

Algorithm 1 Particle Method

- 1: Emit($\omega_p, T_p, \rho_p, \tau$)
 - 2: $\tilde{T}_p, \tilde{\rho}_p \leftarrow \text{Diffuse}(T_p, \rho_p, k, \Delta t)$
 - 3: $F_{bv} = -(\rho_0 / \tilde{\rho}_p)g - \beta(\tilde{T}_p - T_0)g$
 - 4: $u_p \leftarrow \omega_p$
 - 5: $\tilde{\omega}_p \leftarrow \text{Stretch}(\omega_p, u_p, \Delta t)$
 - 6: $\tilde{\omega}_p = \tilde{\omega}_p + \nabla \times F_{bv}$
 - 7: $\omega_p^{n+1} \leftarrow \text{Diffuse}(\tilde{\omega}_p, k, \Delta t)$
 - 8: $p^{n+1} \leftarrow \text{RK3}(p^n, u_p, \Delta t)$
 - 9: if($t == \tau$) Remove
-

The specific steps are as follows:

(1) The system emits a group of particles including vorticity ω_p , temperature T_p , density ρ_p and particle lifetime τ .

(2) Temperature and density diffusion: It is very difficult to accurately simulate the viscous diffusion of particles. But a simple "trick" can be used to demonstrate particle diffusion[29]. Taking temperature as an example, according to the upward ejection of hydrothermal fluid, it can be assumed that the temperature of the smoke ejected from the nozzle is the highest, and when the smoke reaches the neutral buoyancy surface, the temperature is the lowest. Therefore, a function can be constructed that makes the temperature T_p and the temperature decrease continuously over time, that is, $\tilde{T}_p = T_p e^{-kt}$, where k is the coefficient that controls temperature diffusion. As shown in Figure 8, the temperature of each particle T_p and at the initial moment of eruption, its temperature is the maximum value T_p , and then as the particles continue to rise, their temperature continues to decrease. After a certain period of time, its temperature reaches its minimum value. In the same way, a density function can be constructed, that is, $\tilde{\rho}_p = \rho_p e^{-kt}$, where k is the coefficient that controls the density diffusion.

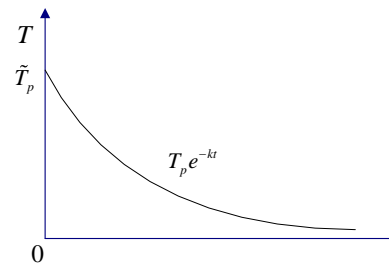


Figure 8. Variation of temperature T_p with time t

(3) According to the obtained new temperature \tilde{T}_p and density $\tilde{\rho}_p$ on the particle, the buoyancy on the grid can be obtained as

$$F_{bv} = -g \frac{\rho_0}{\tilde{\rho}_p} - \beta g (\tilde{T}_p - T_0) \quad (34)$$

(4) We use the Biot-Savart rule to perform the N-body summation. According to the particle velocity equation (25) obtained by analytic discretization, the velocity of any vortex particle can be obtained as

$$u_p = \sum_{j,j \neq p}^N \frac{\omega_p \times (r_p - r_j)}{4\pi |r_p - r_j|^3} \quad (35)$$

Where r_p is the position of the vortex particle, and r_j is the position of any other particle.

(5) According to the equation (31) that uses the particle method to solve the vortex of the stretching term, we convert the vortex particles into the position and velocity of the two ends, respectively, $x_{p,a}$, $x_{p,b}$ and $u_{p,a}$, $u_{p,b}$, the new position of the vortex segment is

$$\begin{aligned} x_{p,a}^{n+1} &= x_{p,a}^n + \Delta t u_{p,a}^n \\ x_{p,b}^{n+1} &= x_{p,b}^n + \Delta t u_{p,b}^n \end{aligned} \quad (36)$$

After the vortex segment is updated, the vortex segment is stretched, and then converted to the vorticity of the new vortex particle as

$$\omega_p = \kappa (x_{p,b}^{n+1} - x_{p,a}^{n+1}) \quad (37)$$

(6) According to the buoyancy of the particle, the new vorticity of the vortex particle can be obtained as

$$\tilde{\omega}_p = \omega_p + \nabla \times F_{bv} \quad (38)$$

(7) Vortex diffusion: Similar to step (2), according to the process of smoke rising, it can be assumed that the vorticity of the first sprayed smoke is the largest. When the smoke rises to a certain position, its velocity and vorticity All tend to 0. Therefore, a function can be constructed that makes the vorticity ω_p decrease continuously over time, that is, $\omega_p^{n+1} = \tilde{\omega}_p e^{-kt}$, where k is the coefficient that controls the vorticity diffusion.

(8) we use the third-order Runge-Kutta to find the position p^{n+1} of the next stage of tracking particles. First, get the particle velocity as

$$k_1 = u(p^n) \quad (39)$$

Iterate this velocity to the next step to obtain the particle velocity

$$k_2 = u(p^n + \frac{1}{2} \Delta k_1) \quad (40)$$

(2) Assign the particle temperature, density and vorticity to the corresponding grid nodes. Divide the grid in the calculation plane. According to the area weighting method (Figure 4), the vorticity ω , temperature T and density ρ at each grid node can be obtained.

$$\begin{aligned}\omega &= \sum_{k=1}^n \omega(k) \\ T &= \sum_{k=1}^n T(k) \\ \rho &= \sum_{k=1}^n \rho(k)\end{aligned}\quad (43)$$

(3) Solve the stream function vector ψ and velocity field u . First set the initial value $\psi^{(0)} = 0$, and then solve the Poisson equation (21). Using the finite difference method in the grid, we can get:

Two-dimensional:

$$\psi_{i,j}^{(n+1)} = (\omega + \psi_{i-1,j}^{(n)} + \psi_{i+1,j}^{(n)} + \psi_{i,j-1}^{(n)} + \psi_{i,j+1}^{(n)})/4 \quad (44)$$

Three-dimensional:

$$\psi_{i,j,k}^{(n+1)} = (\omega + \psi_{i-1,j,k}^{(n)} + \psi_{i+1,j,k}^{(n)} + \psi_{i,j-1,k}^{(n)} + \psi_{i,j+1,k}^{(n)} + \psi_{i,j,k-1}^{(n)} + \psi_{i,j,k+1}^{(n)})/6 \quad (45)$$

Most of vortex methods use fast and effective numerical solutions to elliptic equations. Using the iterative method, starting from an initial "guess" value $\psi^{(0)}$, each step n produces an improved solution $\psi^{(n)}$, gradually approaching the exact value. The superscript symbol indicates the number of iterations. Common solution methods include Jacobian iteration, Gauss-Seidel iteration, conjugate gradient, multigrid method [30], etc. In order to achieve rapid convergence, we adopt the multigrid method.

After obtaining ψ , the value of u can be obtained according to $u = \nabla \times \psi$.

$$u = \nabla \times \psi = \left(\frac{\partial \psi_3}{\partial y} - \frac{\partial \psi_2}{\partial z}, \frac{\partial \psi_1}{\partial z} - \frac{\partial \psi_3}{\partial x}, \frac{\partial \psi_2}{\partial x} - \frac{\partial \psi_1}{\partial y} \right) \quad (46)$$

Where ψ_1 , ψ_2 , and ψ_3 are the three components of ψ . In 2D, because ψ_1 and ψ_2 are 0, then

$$u = \nabla \times \psi = \left(\frac{\partial \psi_3}{\partial y}, -\frac{\partial \psi_3}{\partial x}, 0 \right) \quad (47)$$

(4) This step uses the induced velocity u obtained in step (3) and combines the semi-Lagrangian method to advection temperature, density and vorticity. Taking the temperature solution as an example, Figure 10 shows the temperature field, the temperature on each grid (represented by pure black dots). To calculate the temperature at time t (marked with a red circle), the temperature field needs to be traced back to the red cross \times . Then we perform bilinear interpolation on the 4 temperature scalars (such as red box) at the grid closest to the red cross \times .

The new temperature \tilde{T} can be solved by advection temperature. According to the Equation (13), the position of red \times is defined as p , and any grid point is defined as $p_{i,j,k}$, then $p = p_{i,j,k} - \Delta t u / \Delta h$, where Δh is the grid size in this direction. Then according to the position of p , find the eight grid points adjacent to the position, and then calculate it based on bilinear interpolation (area weighting) (Figure 4)

$$\tilde{T} = \sum_{k=1}^8 T_p S_p(k) \quad (48)$$

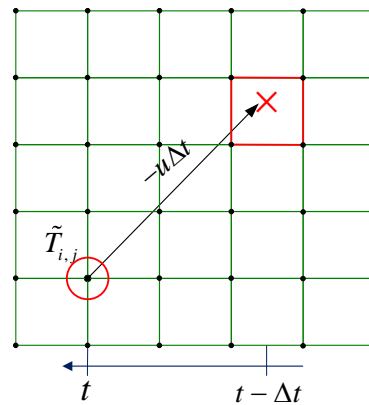


Figure 10. Linear traceback of temperature in grid

Similarly, according to Equation (14) and Equation (15), the density and vorticity are respectively

$$\bar{\rho} = \sum_{k=1}^8 \rho_p S_p(k) \quad (49)$$

$$\bar{\omega} = \sum_{k=1}^8 \omega_p S_p(k) \quad (50)$$

(5) The temperature diffusion is shown in Figure 11:

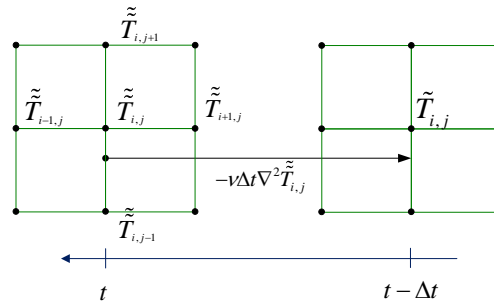


Figure 11. Temperature diffusion

According to Figure 11 and Equation (17), combined with the semi-Lagrangian method, which can be solved iteratively by referring to Equation (21).

$$\tilde{T}_{i,j,k} - \frac{k_T \Delta t}{\Delta h^2} \left(\tilde{T}_{i-1,j,k} + \tilde{T}_{i+1,j,k} + \tilde{T}_{i,j-1,k} + \tilde{T}_{i,j+1,k} + \tilde{T}_{i,j,k-1} + \tilde{T}_{i,j,k+1} - 6\tilde{T}_{i,j,k} \right) = \tilde{T}_{i,j,k} \quad (51)$$

Then

$$\tilde{T}_{i,j,k} = \frac{\tilde{T}_{i,j,k} + \frac{k_T \Delta t}{\Delta h^2} \left(\tilde{T}_{i-1,j,k} + \tilde{T}_{i+1,j,k} + \tilde{T}_{i,j-1,k} + \tilde{T}_{i,j+1,k} + \tilde{T}_{i,j,k-1} + \tilde{T}_{i,j,k+1} \right)}{1 + \frac{6k_T \Delta t}{\Delta h^2}} \quad (52)$$

Similarly, according to the Equation (18) and the Equation (19), the diffusion of the density field and the vortex field can be obtained respectively

$$\tilde{\rho}_{i,j,k} = \frac{\tilde{\rho}_{i,j,k} + \frac{k_\rho \Delta t}{\Delta h^2} \left(\tilde{\rho}_{i-1,j,k} + \tilde{\rho}_{i+1,j,k} + \tilde{\rho}_{i,j-1,k} + \tilde{\rho}_{i,j+1,k} + \tilde{\rho}_{i,j,k-1} + \tilde{\rho}_{i,j,k+1} \right)}{1 + \frac{6k_\rho \Delta t}{\Delta h^2}} \quad (53)$$

$$\tilde{\omega}_{i,j,k} = \frac{\tilde{\omega}_{i,j,k} + \frac{v\Delta t}{\Delta h^2} \left(\tilde{\omega}_{i-1,j,k} + \tilde{\omega}_{i+1,j,k} + \tilde{\omega}_{i,j-1,k} + \tilde{\omega}_{i,j+1,k} + \tilde{\omega}_{i,j,k-1} + \tilde{\omega}_{i,j,k+1} \right)}{1 + \frac{6v\Delta t}{\Delta h^2}} \quad (54)$$

(6) The interaction between the local vorticity vector and the velocity gradient leads to vorticity stretching. In a 3D fluid, when the vortex element passes through advection, it is simultaneously stretched by the velocity field, changing the vorticity field. By separating Euler's equations, need to solve

$$\frac{\partial \omega}{\partial t} = \omega \cdot \nabla u \quad (55)$$

According to Equation (33), define two positions $p1 = p_{ijk} + \omega\Delta t/(2\Delta h)$, $p2 = p_{ijk} - \omega\Delta t/(2\Delta h)$, $p1$ and $p2$ are shown in figure 12 in the vorticity field:

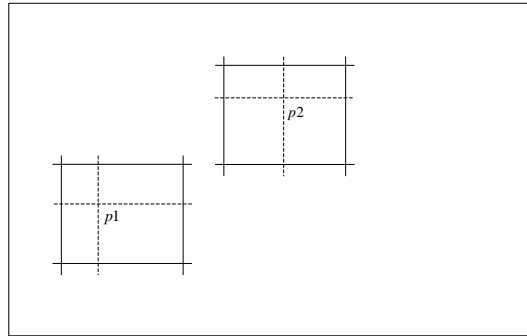


Figure 12. The position distribution of $p1$ and $p2$ in vorticity field in the stretch term

Extended to three-dimensional grid coordinate representation

$$\tilde{\tilde{\omega}} = \tilde{\omega} + u1(p1) - u2(p2) \quad (56)$$

According to Figure 12, combined with the area weighting method, we can get

$$\begin{aligned} u1(p1) &= \sum_{k=1}^8 u_{p1} S_{p1}(k) \\ u2(p2) &= \sum_{k=1}^8 u_{p2} S_{p2}(k) \end{aligned} \quad (57)$$

The new vorticity

$$\tilde{\tilde{\omega}} = \tilde{\omega} + \sum_{k=1}^8 u_{p1} S_{p1}(k) - \sum_{k=1}^8 u_{p2} S_{p2}(k) \quad (58)$$

(7) According to the obtained temperature and density on the new grid, the external influence on the vorticity field can be obtained. Therefore, the new vorticity

$$\tilde{\tilde{\omega}} = \tilde{\tilde{\omega}} + \nabla \times F_{bv} = \tilde{\tilde{\omega}} - \nabla \frac{\rho_0}{\tilde{\rho}} \times g - \nabla \beta (\tilde{T} - T_0) \times g \quad (59)$$

(8) In this step, we need to return to the Lagrangian framework to obtain the motion of the vortex particles, and interpolate the vorticity on the grid to the vortex particles. According to Figure 4, we first divide the vorticity $\tilde{\tilde{\omega}}(k)$ on the grid node by the assigned node weight $W(k)$, Then use the area weighting method to sum the values on the obtained nodes to obtain the vorticity of the vortex particle

$$\omega_p = \sum_{k=0}^8 \frac{\tilde{\tilde{\omega}}(k)}{W(k)} S(k) \quad (60)$$

At the same time, in order to describe the motion trajectory of the hydrothermal fluid, it is necessary to interpolate the velocity on the grid to the tracking particles, and then use the area weighting again according to the velocity $\tilde{u}(k)$ on the grid node that has been calculated Method interpolation to find the velocity of the trajectory particle

$$u_p = \sum_{k=0}^8 \frac{u(k)}{W(k)} S(k) \quad (61)$$

(9) Same as step 8 of Algorithm 1.

(10) If the particle reaches its lifetime τ , remove the particle.

4.3. Hybrid methods (Vortex-In-Cell method)

The processing of temperature and density is mainly on the grid, and the semi-Lagrangian method is used. As mentioned above, this method can achieve unconditional stability. No matter how large the step length is, it does not need to be stepped Stable condition judgment. At the same time, when dealing with vorticity advection, we use the Vortex-In-Cell method. This method saves the vorticity in the form of particles in the vortex particles, avoiding the numerical dissipation caused by the use of grids, and when dealing with induced velocity When the vortex is stretched, it is processed on the grid. This method is simple, efficient, and can fully show the details of fluid changes. The flowchart is shown in Figure 13. Based on this, the specific implementation process is shown in Algorithm 3:

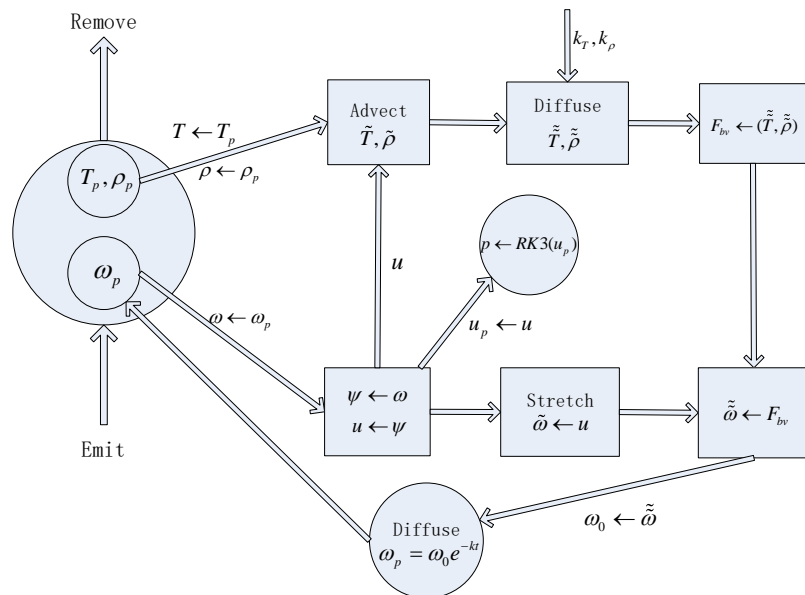


Figure 13. Flow chart of smoke (hybrid method)

Specific steps are as follows:

- (1) The system emits a group of particles including vorticity ω_p , temperature T_p , density ρ_p and particle lifetime τ .
- (2) Same as step 2 of Algorithm 2.
- (3) Same as step 3 of Algorithm 2.
- (4) The advection of temperature and density is the same as step 4 of Algorithm 2.
- (5) The advection of temperature and density is the same as step 5 of Algorithm 2.
- (6) Similar to step 6 of Algorithm 2.
- (7) Similar to step 7 of Algorithm 2.
- (8) Similar to step 8 of Algorithm 2.
- (9) Same as step 7 of Algorithm 1.
- (10) Same as step 8 of Algorithm 1.
- (11) If the particle reaches its lifetime τ , remove the particle.

Algorithm 3 Hybrid Method

```

1: Emit( $\omega_p, T_p, \rho_p, \tau$ )
2:  $\omega \leftarrow \omega_p, \quad T \leftarrow T_p, \quad \rho \leftarrow \rho_p$ 
3:  $\psi \leftarrow \nabla^2 \psi = -\omega, \quad u \leftarrow \nabla \times \psi$ 
4:  $\tilde{T}, \tilde{\rho} \leftarrow \text{Advection}(T, \rho, u, \Delta t)$ 
5:  $\tilde{T}, \tilde{\rho} \leftarrow \text{diffusion}(\tilde{T}, \tilde{\rho}, k_T, k_\rho, \Delta t)$ 
6:  $\tilde{\omega}, \tilde{u} \leftarrow \text{stretch}(\omega, u, \Delta t)$ 
7:  $\tilde{\omega} = \tilde{\omega} + \nabla \times F_{bv} = \tilde{\omega} - \nabla(\rho_0/\tilde{\rho}) \times g - \nabla\beta(\tilde{T} - T_0) \times g$ 
8:  $\omega_0 \leftarrow \tilde{\omega}, \quad u_p \leftarrow \tilde{u}$ 
9:  $\omega_p^{n+1} \leftarrow \text{diffusion}(\omega_p, k, \Delta t)$ 
10:  $p^{n+1} \leftarrow \text{RK3}(p^n, u_p, \Delta t)$ 
11: if( $t == \tau$ ) Remove

```

5. Experiments and conclusions

This experiment uses three modeling methods (particle method, grid method and hybrid method) to simulate smoke. In this experiment, the three methods all use 1 600 vortex particles to provide vorticity and 409 600 tracking particles to track the trajectory; at the same time, in the grid method and the hybrid method, the grid scale is $32 \times 64 \times 32$. The final real scene and the rendering scene of the three methods are shown in Figure 14.

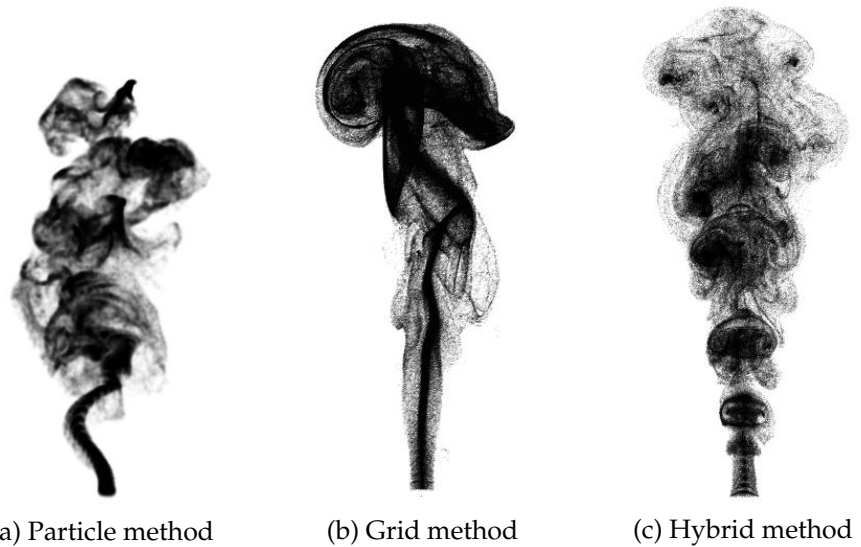


Figure 14. Three methods of smoke simulation

In Figure 14(a), although the particle method avoids the problem of numerical dissipation caused by grid calculation, the fluid trajectory is distorted and appears unreal. In Figure 14(b), the approximate outline of the smoke is drawn using the grid method, but due to the existence of numerical dissipation, numerical viscosity is generated, making the surface relatively smooth. In Figure 14(c), the vorticity is stored in the form of particles using a hybrid method, which avoids the problem of numerical dissipation caused by the grid, and when dealing with other variables (such as velocity, vortex stretching, etc.), the grid method is used, which fully shows the details of turbulence.

Funding: This research was funded by the Research Startup Fund of Henan University of Urban Construction [No. K-Q2020025].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Foster, N.; Metaxas, D. Modeling the Motion of a Hot, Turbulent Gas. *ACM SIGGRAPH*, 1997, pp. 181–188.
2. Stam, J. Stable fluids. *ACM SIGGRAPH*, 1999, pp. 121–128.
3. Fedkiw, R.; Stam, J.; Jensen, H.W. Visual simulation of smoke. *ACM SIGGRAPH*, 2001, pp. 15–22.
4. Kim, B.; Liu, Y.; Llamas, I.; Rossignac, J. Flow-Fixer: Using BFECC for fluid simulation. First Eurographics Conf. on Natural Phenomena, NPH05, 2005, pp. 51–56.
5. Selle, A.; Fedkiw, R.; Kim, B.; Liu, Y.; Rossignac, J. An Unconditionally Stable MacCormack Method. *Scientific Computing* **2008**, 35, 350–371.
6. Zhu, Y.; Bridson, R. Animating Sand as a Fluid. *ACM Trans. Graph.* **2005**, 24, 965–972.
7. Barnes, J.; Hut, P. A Hierarchical O(N log N) force calculation algorithm. *Nature* **1986**, 324, 446–449.
8. Greengard, L. *The Rapid Evaluation of Potential Fields in Particle Systems*; ACM Press, 1987.
9. Christiansen, J.P. Numerical simulation of hydrodynamics by the method of point vortices. *J. Comp. Phys.* **1973**, 13, 363–379.
10. Harlow, F.H.; Welch, J.E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Physics of fluids* **1965**, 8, 2182–2189.
11. Selle, A.; Rasmussen, N.; Fedkiw, R. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.(Proc. SIGGRAPH)* **2005**, 24, 910–914.
12. Pfaff, T.; Thuerey, N.; Cohen, J.; Tariq, S.; Gross, M. Scalable fluid simulation using anisotropic turbulence particles; *ACM SIGGRAPH Asia*, 2010; pp. 174:1–8.
13. Weißmann, S.; Pinkall, U. Filament-based smoke with vortex shedding and variational reconnection **2010**. 29.
14. Brochu, T.; Keeler, T.; Bridson, R. Lineartime smoke animation with vortex sheet meshes. *Proc. Symp. Comp. Animation*, 2012, pp. 87–95.
15. Park, S.I.; Kim, M.J. Vortex fluid for gaseous phenomena. *Proc. Symp. Comp. Animation*, 2005, pp. 261–270.
16. Bridson, R.; Hourihane, J.; Nordenstam, M. Curl-noise for procedural fluid flow. *ACM Trans. Graph.(Proc. SIGGRAPH)* **2007**, 26.
17. Sawyer, J.S. A semi-Lagrangian method of solving the vorticity advection equation. *Tellus* **1963**, 15, 336–342.
18. Dritschel, D.G.; Ambaum, M.H. A contour-advective semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields. *Q. J. R. Meteorol. Soc.* **1997**, 123, 1097–1130.
19. Kotelnikov, A.D.; Zabusky, N.J. Vortex dynamics of a twice-accelerated interface in an incompressible ideal fluid. *Astrophysical Journal Supplement Series* **2000**, 127, 389–394.
20. Zhang, X.; Bridson, R. A PPPM fast summation method for fluids and beyond. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* **2014**, 33, 206:1–11.
21. Marshall, J.S.; Grant, J.R. Penetration of a blade into a vortex core: vorticity response and unsteady blade forces. *Fluid Mech.* **1996**, 306, 83–109.
22. Liu, C.H.; Doorly, D.J. Vortex particle-in-cell method for three-dimensional viscous unbounded flow computations. *Num. Meth. Fluids* **2000**, 32, 29–50.
23. Zhang, X.; Bridson, R.; Greif, C. Restoring the missing vorticity in advection-projection fluid solvers. *ACM Trans. Graph.* **2015**, 34, 52:1–8.
24. Graham, J.M.R. Computation of viscous separated flow using a particle method, *Numerical methods for fluid dynamics III*. Oxford, England, 1988, pp. 21–24.
25. Cheng, M.; Chew, Y.T.; Luo, S.C. A hybrid vortex method for flows over a bluff body. *Num. Meth. Fluids* **1997**, 24, 253–274.
26. Lu, Z.Y.; Ross, T. Diffusing-vortex numerical scheme for solving incompressible Navier-Stokes equation. *Comput. Phys.* **1991**, 95, 400–435.
27. Boussinesq, J. *Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes a grande section*; Gauthier-Villars, 1897.
28. Bjerknes, V. Über die bildung cirkulationsebewegungen und wirbeln in rebungslosen flüssigkeiten. *Videnskabselskabet's Skrifter. I Math. Naturv. Klasse* **1898**, 5, 1–29.
29. Bridson, R. *Fluid Simulation for Computer Graphics, Second Edition*; CRC Press, 2015.
30. McAdams, A.; Sifakis, E.; Teran, J. A parallel multigrid poisson solver for fluids simulation on large grids. *ACM SIGGRAPH / Eurographics Symp. Comp. Anim.*, 2010, pp. 65–74.

