

## Reducing numerical dissipation in smoke simulation <sup>☆</sup>



Zhanpeng Huang <sup>a,\*</sup>, Ladislav Kavan <sup>b</sup>, Weikai Li <sup>c</sup>, Pan Hui <sup>c</sup>, Guanghong Gong <sup>a</sup>

<sup>a</sup> Department of Automation Science and Electrical Engineering, Beihang University, Beijing, China

<sup>b</sup> Computer and Information Sciences Department, University of Pennsylvania, PA, USA

<sup>c</sup> Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong

### ARTICLE INFO

#### Article history:

Received 3 March 2014

Received in revised form 18 November 2014

Accepted 28 December 2014

Available online 7 January 2015

#### Keywords:

Smoke and gaseous phenomena

Physically based methods

Numerical scheme

Numerical dissipation

### ABSTRACT

Numerical dissipation acts as artificial viscosity to make smoke viscous. Reducing numerical dissipation is able to recover visual details smeared out by the numerical dissipation. Great efforts have been devoted to suppress the numerical dissipation in smoke simulation in the past few years. In this paper we investigate methods of combating the numerical dissipation. We describe visual consequences of the numerical dissipation and explore sources that introduce the numerical dissipation into course of smoke simulation. Methods are investigated from various aspects including grid variation, high-order advection, sub-grid compensation, invariant conservation, and particle-based improvement, followed by discussion and comparison in terms of visual quality, computational overhead, ease of implementation, adaptivity, and scalability, which leads to their different applicability to various application scenarios.

© 2015 Elsevier Inc. All rights reserved.

## 1. Introduction

Smoke is desirable in visual effect and video game industries. It is also one of challenging problems in computer graphics due to its complexity and turbulence. To obtain realistic smoke and gaseous phenomena, physically based methods with Navier–Stoke Equations (NSEs) have been explored to model underlying fluid dynamics. Although numerically integrating NSEs have been studied in computational fluid dynamics (CFD), computer graphics researches focus on simplified discretization and numerical schemes when visual quality matters most. Simplifications make physically based methods possible for smoke simulation but introduce the numerical dissipation. The numerical dissipation increases fluid viscosity to make it appear more viscous than intended. It degrades the visual

appearance by smearing out fine details and damping down the motion quickly. The numerical dissipation has been recognized to have substantial visual consequences to the smoke simulation.

Many sources introduce numerical dissipation to the course of the smoke simulation. Coarse spatiotemporal discretization produces numerical truncation errors, which is proven to have a form of viscosity [1]. As fluid quantities are only defined on discrete locations such as grid points and particles, interpolation schemes are required to calculate values at undefined positions, which is equivalent to smoothing operations that produce the numerical dissipation. The semi-Lagrangian method [2] is widely used for the smoke simulation attributed to its unconditional stability and ease of implementation, but it generates a large amount of the numerical dissipation in backward tracing and advection subroutines. Many advanced methods are constructed based on the semi-Lagrangian method to guarantee the unconditional stability. However, they also inherit the disadvantage of massive numerical dissipation.

<sup>☆</sup> This paper has been recommended for acceptance by Peter Lindstrom.

\* Corresponding author.

Massive effort has been devoted to combat the numerical dissipation from different aspects. Some methods are developed to eliminate sources of the numerical dissipation. For instance, it is straightforward to reduce the numerical dissipation by increasing spatial resolution and reducing time step. However, both approaches increase computational overhead. Adaptive mesh [3], irregular mesh [4,5], and dynamical mesh [6] are proposed to reduce the numerical dissipation without significantly increasing computation. Rather than directly reducing the numerical dissipation, several methods generate artificial details to compensate for visual loss using vorticity confinement [7,8] and subscale turbulence models [9,10]. Grid-based methods require resampling flow field, which is equivalent to the low-pass filter to smear out high-frequency components. Particle-based methods only carry quantity but do not dissipate quantity, which does not suffer from the numerical dissipation problem. However, particle methods have problems such as particle redistribution. Hybrid particle and grid methods [11,12] are proposed to leverage advantages of particle and grid to reduce numerical dissipation.

In this paper, we investigate the numerical dissipation in smoke simulation in terms of where it comes out, what impact it has, and how to combat it. The rest of paper is organized as follows: we give a brief introduction to the basic smoke simulation in Section 2 and address the sources of the numerical dissipation in Section 3; in Section 4 we investigate and compare methods of combating numerical dissipation from different aspects, following by a conclusion in Section 5.

## 2. Background

Smoke and other gaseous phenomena are normally simplified to be incompressible and homogenous, which does not decrease the applicability to model basic dynamical mechanisms. The NSEs to model smoke are derived as:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{\nabla p}{\rho} + \mathbf{f} & \text{(a)} \\ \nabla \cdot \mathbf{u} = 0 & \text{(b)} \end{cases} \quad (1)$$

where  $\mathbf{u}$  is velocity,  $p$  and  $\rho$  denote pressure and fluid density respectively.  $\nu$  is kinematic viscosity to measure how viscous the fluid is and  $\mathbf{f}$  represents the resultant external force. The two equations indicate that the fluid should conserve both momentum and mass. The first equation is derived from Newton's second law with left-hand term presenting acceleration and right-hand terms the net force exerted on fluid.

NSEs are too complicated to solve for analytical solution directly. The NSEs usually break down into simple terms including advection, pressure, diffusion, and external force [2]. The simple terms can then be easily solved individually. If we define the terms as operators denoted by  $\mathbb{A}$ ,  $\mathbb{P}$ ,  $\mathbb{D}$ , and  $\mathbb{F}$ , the operator  $\mathbb{S}$  to solve NSEs can be written as [13]:

$$\mathbb{S} = \mathbb{P} \circ \mathbb{F} \circ \mathbb{D} \circ \mathbb{A} \quad (2)$$

where

$$\begin{cases} \mathbb{A} : \frac{\partial q}{\partial t} = -(\mathbf{u} \cdot \nabla)q & \text{(a)} \\ \mathbb{D} : \frac{\partial \mathbf{u}}{\partial t} = \nu \nabla^2 \mathbf{u} & \text{(b)} \\ \mathbb{F} : \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f} & \text{(c)} \\ \mathbb{P} : \frac{\partial \mathbf{u}}{\partial t} + \frac{\nabla p}{\rho} = 0, \text{ so that } \nabla \cdot \mathbf{u} = 0 & \text{(d)} \end{cases} \quad (3)$$

where  $q$  can be velocity, temperature, or any other fluid quantity.

## 3. Numerical dissipation as artificial viscosity

Numerical solutions are different from exact solution due to numerical truncation errors. The truncation errors include additional high-order terms which influence fluid motion and appearance. We start with the simple one-dimensional advection to analyze the impact on fluid motion:

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} = 0, \quad u > 0 \quad (4)$$

If we discretize it using forward Euler for the time derivative and first-order backward difference for the space derivative we can get:

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + u \frac{q_i^n - q_{i-1}^n}{\Delta x} = 0 \quad (5)$$

We can rearrange it to get

$$q_i^{n+1} = q_i^n - \Delta t \frac{q_i^n - q_{i-1}^n}{\Delta x} u \quad (6)$$

Recalling the Taylor series for  $q_{i-1}^n$  gives

$$q_{i-1}^n = q_i^n - \left(\frac{\partial q}{\partial x}\right)_i \Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i \frac{\Delta x^2}{2} + O(\Delta x^3) \quad (7)$$

Substituting it into above equation and doing the cancellation gives

$$q_i^{n+1} = q_i^n - \Delta x \left(\frac{\partial q}{\partial x}\right)_i u + \Delta t \Delta x \left(\frac{\partial^2 q}{\partial x^2}\right)_i u + O(\Delta x^2) \quad (8)$$

Deleting the second-order truncation error and rewriting it gets

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + u \left(\frac{\partial q}{\partial x}\right)_i = \Delta x \left(\frac{\partial^2 q}{\partial x^2}\right)_i u \quad (9)$$

Which is the forward Euler in time applied to the modified PDE

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} = u \Delta x \frac{\partial^2 q}{\partial x^2} \quad (10)$$

The Laplacian of  $q$  in one dimension is  $\nabla^2 q = \partial^2 q / \partial x^2$ . Defining  $v' = u \Delta x$  and substituting it into the equation gives

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} = v' \nabla^2 q \quad (11)$$

The modified advection equation includes an additional viscosity-like term. It is called artificial viscosity as it is purely of numerical origin without any physical meanings. Even though the fluid is inviscid, the artificial viscosity performs like physical viscosity to make the fluid viscous.

#### 4. Combating numerical dissipation

##### 4.1. Grid variation

The aforementioned artificial viscosity indicates the numerical dissipation is proportional to grid spacing  $\Delta x$ . In theory, we could reduce the numerical dissipation by refining the grid spacing. However, refining grid dramatically increases computational cost. Taking the widely used MAC grid for example, if the grid resolution increases by  $\gamma$  times in three dimensions, the number of grid cells increases up to  $\gamma^3$  times. As the grid spacing decreases by  $\gamma$ , time step has to reduce to  $1/\gamma$  to meet Courant–Friedrichs–Lewy (CFL) condition (even for the unconditionally stable semi-Lagrangian method, the time step is usually set to be several times of CFL number in practice). The total computational cost rises up to  $\gamma^4$  times, which makes the grid refinement method scale poorly and ill-suited for practical use.

To alleviate the problem, Berger and Oliger [14,15] introduced adaptive mesh refinement (AMR) into CFD to discretize local regions with different grid resolutions. Losasso et al. [3] proposed an octree structure to adaptively resolve irregular boundaries with high-resolution grid while using low-resolution grid at empty space. The method dramatically improves visual effects around irregular boundaries with a small increment of computational cost. Fig. 1 shows visual comparison between the basic method and the octree-based method with similar grid resolutions. As the grid is adaptively refined around irregular boundary, the method captures more visual details than traditional uniform grid method. However, it increases the complexity of grid structure. The additional computational cost pays off only when grid refinements are limited to a few local regions. In addition, non-uniform grid constructs a nonsymmetric linear system to solve for pressure, which may cause potential numerical instability.

As hexahedral grid does not align to horizontal and vertical boundaries, it cannot resolve irregular boundaries

that do not align to coordinate axes. Tetrahedral grid is proposed to represent irregular boundaries with much less numerical errors. Elcott et al. [4] discretized and solved NSEs on tetrahedral grid. The method can present complicated boundaries but has problem of energy dissipation, which can be modified using implicit time-reversible velocity integration on the tetrahedral grid [16]. Tetrahedral grid has intrinsic complicated structure to increase the computational cost. A hybrid method is proposed to combine the hexahedral and tetrahedral structures [5]. The tetrahedral grid is used to resolve irregular boundaries while the hexahedral grid is used at empty regions without obstacles. Fig. 2 gives visual comparison of smoke simulations using different discretization schemes. As most interesting features are generated around irregular boundaries, the hybrid grid is able to obtain similar visual quality as tetrahedral method near boundaries. As the hybrid grid uses uniform grid mesh in most empty spaces, it reduces the total computational cost comparable with the uniform grid method.

Instead of the static grid structure, several works reduce computational overhead by translating and reconstructing grid mesh according to boundary conditions. Shah et al. [6] tracked smoke movement with movable and scalable grid mesh. Empty regions without smoke are not discretized in order to decrease computational cost. To guarantee visual consistence between simulation domain and empty regions, Cohen et al. [17] used a simple particle system outside of the simulation domain. Velocity at domain boundary is assigned as initial conditions of the simple particle system. Zhu et al. [18] extended the uniform grid to create a large far-field grid. The uniform grid maintains fine resolution to capture detailed features in interesting regions while extended coarse grid is used to obtain large field view of regions that does not require fine presentation. Fig. 3 compares three different grid configurations. In left figure, the grid is translated along bulk motion of smoke. Only regions filled with smoke are computed to reduce computational overhead. The smoke region expands as smoke moves around, which can be partitioned into two subregions as illustrated in the middle figure. The smoke subregion is calculated using physical methods to capture major body of smoke while the less important subregion is modeled with simple non-physical methods to reduce computation. In order to simulate smoke from a



Fig. 1. Visual comparison of smoke simulations using octree grid (left) [3] structure and MAC (right) [7].

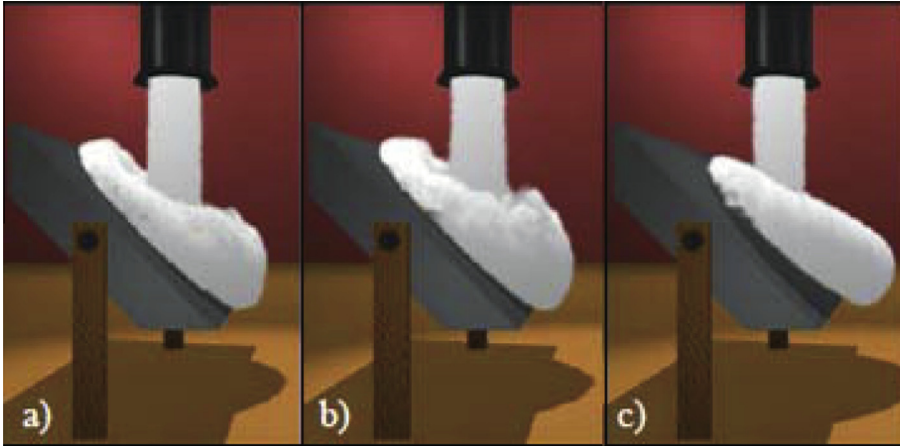


Fig. 2. Smoke simulations using (a) tetrahedra mesh, (b) hybrid mesh, and (c) hexahedra mesh [5].

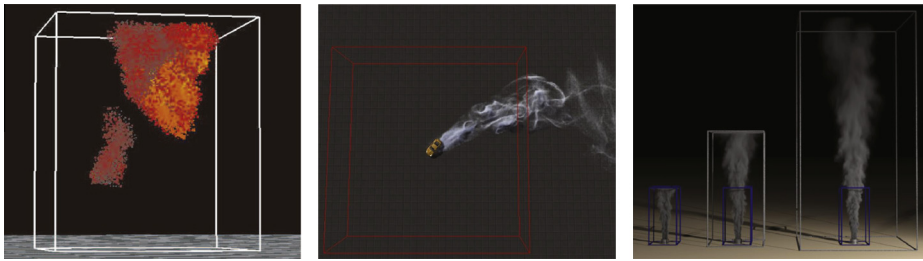


Fig. 3. Left: The fully bounded translating grid [6]; middle: partially bounded translating grid [17]; right: fully bounded translating grid with different resolutions [18].

far distance, the right figure extends coarse cells around fine grid to obtain a large-field view while maintaining fine resolution at local regions of interest. Other than translating and extending the grid mesh, Feldman et al. [19] and Klingner et al. [20] dynamically deformed tetrahedral grid mesh to adapt to variable boundaries and moving obstacles. All adaptive grid methods have common drawbacks of high complicated structure and high computational cost.

A full description of truncated error includes both  $\Delta x$  and  $\Delta t$  terms. Reducing time step also reduces numerical dissipation at the cost of increasing computational overhead. In smoke simulation, we have to carefully choose the time step that does not produce strange results and breaks the numerical stability.

#### 4.2. Advection improvement

The semi-Lagrangian advection is predominant attributed to its simplicity and unconditional stability. The method is widely used as the basic blocks to construct other advanced advection schemes. The basic semi-Lagrangian advection is given as:

$$\mathbf{u}'(\mathbf{x}, t + \Delta t) = \mathbf{u}^n(\mathbf{x} - \mathbf{u}^n(\mathbf{x}, t)\Delta t, t) \quad (12)$$

which means that the velocity at location  $\mathbf{x}$  at next time step is the velocity at location  $(\mathbf{x} - \mathbf{u}^n(\mathbf{x}, t)\Delta t)$  at current time step  $t$ . As current velocity is always bounded to previous velocities, it guarantees unconditional stability.

#### 4.2.1. High-order backward tracing

To get the previous position where current quantity ends up at inquired grid point at the next time step, the basic advection employs the forward Euler scheme to trace the trajectory in reverse to calculate previous position, leading to certain numerical dissipation. Some high-order schemes such as Runge–Kutta can be used to improve

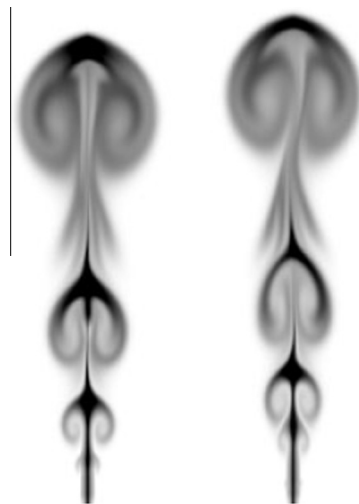


Fig. 4. 2D smoke simulations using the semi-Lagrangian advection coupled with the forward Euler (left) and the second-order Runge–Kutta (right) methods.

the accuracy. Bridson [1] recommended a second-order Runge–Kutta for backward position tracing. Fig. 4 shows the visual results using the first-order forward Euler and the second-order Runge–Kutta. The Runge–Kutta scheme produces more curly details at local regions.

The backward tracing only introduces a fractional part of numerical dissipation to the entire simulation. Boosting accuracy of the backward tracing with high-order schemes limits visual improvement to the final results. In addition, the high-order schemes require more computational cost. The second-order Runge–Kutta is regarded as a default scheme to balance numerical accuracy and computational cost.

#### 4.2.2. High-order interpolation

Most likely the previous position is not on grid points where quantities are defined and stored. We have to interpolate nearby grid points to obtain a good approximation. The linear interpolation is predominant due to its simplicity and stability, but it suffers from massive numerical dissipation because the linear interpolation is similar to the low-pass filter. Catmull–Rom interpolation has second-order numerical accuracy, but it likely overshoots locally and drops into an unstable feedback loop to make simulation blow up finally. Simply preventing overshoot produces additional numerical dissipation in the vicinity of local minima and maxima [1]. Fedkiw et al. [7] clamped slopes to zero where slope signs are flipped. The modified scheme is locally monotone to guarantee stability and second-order accuracy. However, the method is excessive since it flattens overshoots which may be just fine. Huang et al. [21] proposed to adaptively flatten overshoots that break global bounds in order to maintain diversity of fluid features. As it does not require full-scale suppression, the scheme also reduces computational overhead. A visual comparison of smoke using linear, modified, and adaptive Catmull–Rom interpolations is shown in Fig. 5.

There are numerous high-order interpolations in CFD [22], but they are not widely used in smoke simulation for several reasons. As all quantities on grid points require interpolation, a small computational increment of interpolation will significantly increase total computational overhead. In [7], time cost of the monotonic scheme is about 18 times larger than the linear interpolation. Even though the adaptive scheme reduces computational cost, it is about 3–4 times larger than the linear scheme [21]. In addition, the high-order interpolations require more points to construct stencil, which makes it difficult to handle inner boundary conditions. Wide stencil is also complicated to

handle adaptive and non-uniform grids. High-order interpolations require limiters to avoid oscillations, new extrema, and possible instability [23]. Similar to backward tracing, interpolation is not the major contribution to the total numerical dissipation, but it likely increases total computational overhead. We have to choose interpolation scheme based on numerical accuracy versus runtime performance trade-off curve.

#### 4.2.3. High-order advection

The basic semi-Lagrangian advection is popular for smoke simulation. However, excessive numerical dissipation makes it ill-suited for simulation that requires high visual quality. Kim et al. [24,25] applied the Back and Forth Error Compensation and Correction (BFEC) to the semi-Lagrangian Courant–Isaacson–Rees (CIR) to obtain second-order accuracy both in time and space. Forward and backward semi-Lagrangian schemes are compared to estimate numerical error of a single semi-Lagrangian advection, which is then used to compensate a third forward semi-Lagrangian advection. Fig. 6 shows visual comparisons using the semi-Lagrangian and the BFEC advectations. The BFEC generates much more detailed features that the basic semi-Lagrangian scheme, especially vortical features that are swept out by the numerical dissipation.

The backward advection implies that the equation is able to evolve backward, which is inappropriate for parabolic and non-reversible partial differential equations. The BFEC requires great computational cost as it calculates the semi-Lagrangian advection three times. Selle et al. [23] proposed a modified MacCormack scheme which requires much less computational cost. In the BFEC method, the error is treated as a current time quantity so that it is advected forward to the next time step for compensation. Since no strong proof shows that it is a quantity related to the current step, the error can also be used to correct the first forward semi-Lagrangian advection. Fig. 7 illustrates the visual comparison of smoke using the basic semi-Lagrangian and the modified MacCormack methods. The MacCormack scheme generates more details than the semi-Lagrangian method. Comparing visual improvements using high-order interpolations and integrators in Figs. 4 and 5, boosting numerical accuracy of advection schemes is able to obtain greater improvement. It also reduces computational overhead as it does not require a third advection.

Since the BFEC and the modified MacCormack are both constructed based on the semi-Lagrangian scheme, the two



Fig. 5. Smoke ball collision simulations using linear scheme (left), mono Catmull–Rom (middle), and adaptive Catmull–Rom interpolations (right) [21].

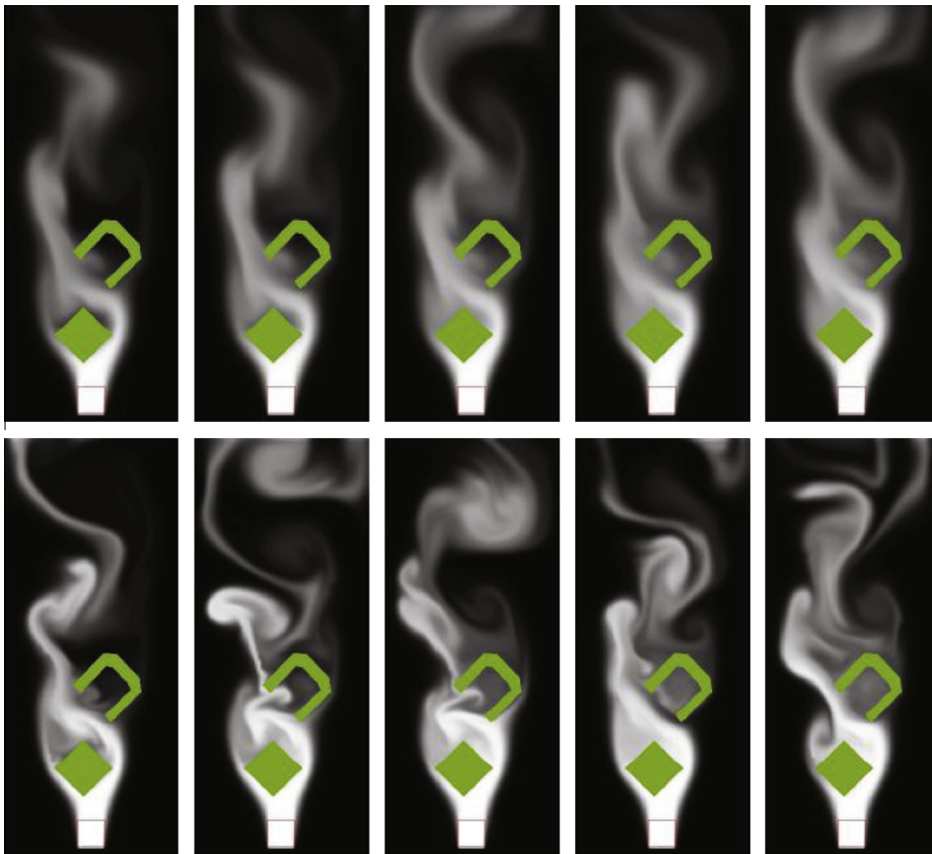


Fig. 6. Smoke simulations with the semi-Lagrangian (top) and the BFECC methods (bottom) [25].



Fig. 7. Smoke simulations with the semi-Lagrangian (left) and the modified MacCormack methods (right) [17].

methods are unconditionally stable. However, the back and forth round step makes them inappropriate to work near interface of multiphase flows. It may not conserve momentum if the first forward advection goes across to the other side fluid with different density and then comes back with then same velocity. The same problem happens near boundaries of obstacles and computational domain. It is suitable to replace them with the semi-Lagrangian scheme near these regions. In the semi-Lagrangian method time step is normally set to be less than 5 times of CFL condition number, so it is safe to turn off the BEFCC and the

modified MacCormack at  $5\Delta x$  off the interface and boundaries.

Other than advecting velocity, Molemaker et al. [26] proposed the QUICK scheme to advect velocity fluxes. QUICK [27] is a third-order accurate explicit scheme. It produces numerical dissipation only at finest spatial scale that can be resolved by the grid. Fig. 8 illustrates the visual results with different advectations. The QUICK scheme generates much more details than the basic semi-Lagrangian method, especially on high-resolution grid.

The first-order forward Euler is unstable for the QUICK [28]. It can be replaced with other temporal schemes with large stability domains such as second-order Adams–Bashforth (AB2) [29]. As the QUICK is conditionally stable only when it meets CFL condition, it greatly increases computational cost. Attributed to its regular memory access pattern, it normally requires less computational cost than other semi-Lagrangian based schemes with careful memory access.

Apart from advecting velocity flux, several methods use constrained interpolation profile (CIP) [30,31] to advect both the velocity and its derivatives, which can boost numerical accuracy to third order. However, original CIP requires a compact stencil and it is conditionally stable with tight CFL condition. The method does not consider derivatives at farthest cell corner, so it only guarantees numerical accuracy when backward-traced point is not far from the start corner. To improve the numerical stability, Song et al. [32] proposed the monotonic CIP (MCIP) by suppressing potential overshoots along all dimensions. The excessive suppression obtains unconditional stability but introduces some numerical dissipation. The dimension-splitting strategy also dramatically increases computational cost [33]. As splitting computation is related to axis directions, it normally produces some numerical dissipation. The problem is even serious in shear motions [34]. Kim et al. [35] proposed an unsplit semi-Lagrangian CIP (USCIP) to guarantee unconditional stability. Two additional symmetrical terms are added to consider off-axis movements such as shear and rotation to avoid unnecessary wiggles. It is able to reduce the numerical dissipation with relative low computational overhead. Fig. 9 compares visual results by using various advection schemes. The BFEC and the MCIP obtain more compelling visual results than the basic semi-lagrangian schemes, while the USCIP method produces higher visual quality than other methods. CIP-based schemes advect both velocity and derivatives. Their computational cost is comparable with the BFEC method, but much higher than the basic semi-Lagrangian method. As USCIP does not require dimension splitting, it requires less computation than the BFEC and the USCIP methods.

To illustrate how the numerical dissipation disperses and smears out details, we compared several unconditional stable advectations using the standard Zalesak disk test [36]. The benchmark is a  $100 \times 100$  grid with a slotted disk centered at (50, 75) of 15 cells in radius, 5 cells in width, 25 cells in length of slot and 5 cells of an upper bridge. A constant vorticity velocity field is given as [37]:

$$\begin{cases} u = (\pi/314)(50 - y) & \text{(a)} \\ v = (\pi/314)(x - 50) & \text{(b)} \end{cases} \quad (13)$$

so that the slotted disk finishes one revolution every 628 time steps.

Fig. 10 shows results of one revolution with different advection schemes. An accurate advection scheme is able to translate and rotate interface without distortion and deterioration, especially at interface with high curvature. The basic semi-Lagrangian method has too much numerical dissipation to maintain basic shape of the disk. The BFEC and MacCormack schemes keep generally shape but smooth slots and corners. The MCIP scheme conserves the disk shape with certain erosion at inner and outer corners, while USCIP almost suffers no distortion and area loss.

Table 1 lists area loss of the disk with different schemes and grid configurations. The “loss” column represents percentage of the area loss. Negative value means the final disk is smaller than the original disk while positive value indicates the disk grows. With the same grid configuration, high-order schemes such as the USCIP suffer from much less area loss than low-order schemes such as the semi-Lagrangian. With the same advection, the area loss also decreases as grid resolution increases. As high-order schemes are able to suppress numerical dissipation even at relative low-resolution grids, the improvement gain is not significant by increasing grid resolutions when using high-order schemes. While first-order semi-Lagrangian method has too much numerical dissipation on coarse grid, using fine grid significantly reduces total numerical dissipation. We can also find that the amount of loss with the semi-Lagrangian method drops relatively much faster than that of high-order schemes, which indicates that low-order schemes are much more sensitive to the grid resolution than high-order schemes. Table 2 compares discussed advectations in terms of numerical stability, numerical accuracy, and computational cost.

Since the backward tracing and interpolation schemes are orthogonal, aforementioned high accurate tracking schemes in Section 4.2.1, high-order interpolations in Section 4.2.2 and high-order advectations in Section 4.2.3 can be integrated together to further improve the numerical accuracy. In addition, other smoke quantities including density, temperature, and concentration can also be advected with above methods to reduce total numerical dissipation. However, if high-order schemes are excessively used for all quantities, high-frequency components of quantities will accumulate simultaneously to produce highly sharp



Fig. 8. Simulations using the semi-Lagrangian advection (left) and the QUICK advectations with low- and high-resolution grid configurations [26].

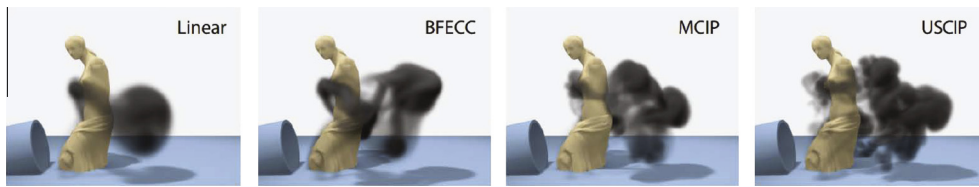


Fig. 9. Smoke simulations with various advection schemes [35].

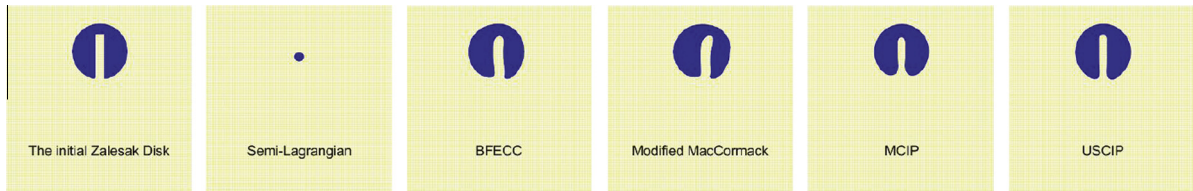


Fig. 10. Zalesak disk after one revolution advected by different schemes [38].

Table 1

Area loss of disk with different advection and grid configurations (initial area =  $5.82e-2$ ).

Size	Semi-Lag.		MacCormack		BFEC		QUICK		USCIP	
	Area	Loss (%)	Area	Loss (%)	Area	Loss (%)	Area	Loss (%)	Area	Loss (%)
$100^2$	$4.20e-3$	-92.8	$5.75e-2$	-1.37	$5.99e-2$	2.92	$5.68e-2$	-2.37	$5.71e-2$	-1.87
$200^2$	$3.97e-2$	-31.5	$5.78e-2$	-0.712	$5.76e-2$	-1.03	$5.84e-2$	0.345	$5.80e-2$	-0.376
$400^2$	$5.34e-2$	-8.25	$5.79e-2$	-0.408	$5.83e-2$	0.247	$5.83e-2$	0.171	$5.826e-2$	0.118
$800^2$	$5.43e-2$	-6.70	$5.81e-2$	-0.201	$5.818e-2$	-0.04	$5.819e-2$	-0.017	$5.822e-2$	0.03

Table 2

A list of several advectons for smoke simulation.

Advection	Numerical stability	Numerical accuracy	Computational cost	Representative reference
Semi-Lagrangian	Unconditional stable	First-order	Low	[2]
BFEC	Unconditional stable	Second-order	High	[24]
MacCormack	Unconditional stable	Second-order	Medium	[23]
QUICK	Conditional stable	Third-order	High	[26]
MCIP	Unconditional stable	Third-order	High	[32]
USCIP	Unconditional stable	Third-order	Medium	[35]

interface. For instance, Fig. 11 shows two 2D smoke simulations using different advection configurations. Left image shows result using a combination of the modified MacCormack scheme for velocity and the semi-Lagrangian for density; right image uses the modified MacCormack scheme for both velocity and density. Although the right one has much less numerical dissipation for velocity and density, they accumulate to produce clear sharp interface, which makes it artificial against the real smoke.

### 4.3. Sub-grid compensation

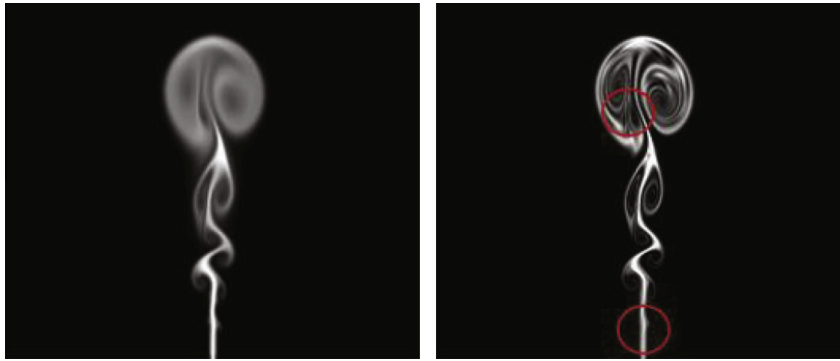
The numerical dissipation increases the viscosity to make smoke inadequate of highly energetic and turbulent appearance. Rather than directly reducing the numerical dissipation by changing discretization schemes and simulation algorithms, many works focus on how to add small-scale features back with heuristic models.

#### 4.3.1. Vorticity confinement

Fedkiw et al. [7] introduced the vorticity confinement method [39] into smoke simulation. The method injects back a certain amount of energy to recover turbulent features by adding a tuneable body force. The force is perpendicular to the direction of vorticity maximum magnitude to enhance local rotation. It is designed to be linear proportional to the grid spacing so that the modified NSEs degenerate to the original NSEs as the additional term vanishes in the limit. Selle et al. [11] coupled the vorticity confinement with vorticity equation. The vorticity confinement force can be synthesized directly from current vorticity field. The left two images of Fig. 12 compare visual results using the basic and the vorticity confinement methods. Injecting some vortical force back into flow significantly improves visual effects, but it also introduces some random artificial turbulence at local regions.

The vorticity confinement is an inexpensive approach to recover small features eliminated by the numerical





**Fig. 11.** Using high-order schemes to advect all quantities produces sharp interface. Left simulation advects velocity with the MacCormack and density with the semi-Lagrangian scheme; right one advects both velocity and density with the MacCormack scheme.

dissipation. However, the method cannot determine how much energy should be injected back without causing instability. Fig. 12 gives comparisons by injecting different amount of vorticity confinement forces. The simulation becomes quasi-unstable and degenerates to a random turbulent chaos when too much force is added. The problem lies in the fact that the coefficient is constant over the entire simulation domain. To solve the problem, He et al. [8] presented an adaptive vorticity confinement coefficient according to helicity other than user-defined constant value. Fig. 13 illustrates the visual comparison between the constant and variable vorticity confinement methods. They further developed a robust second vorticity confinement method [40] to guarantee the numerical stability even with large confinement coefficients.

Vorticity confinement methods add kinetic energy to dilute energy diffusion caused by the numerical dissipation, but they cannot fully compensate for excessive numerical dissipation [26]. It is inappropriate for low viscous fluid that inertia is so well conserved that fluid motion can propagate for a significant distance without being damped and broken down.

#### 4.3.2. Subscale turbulence model

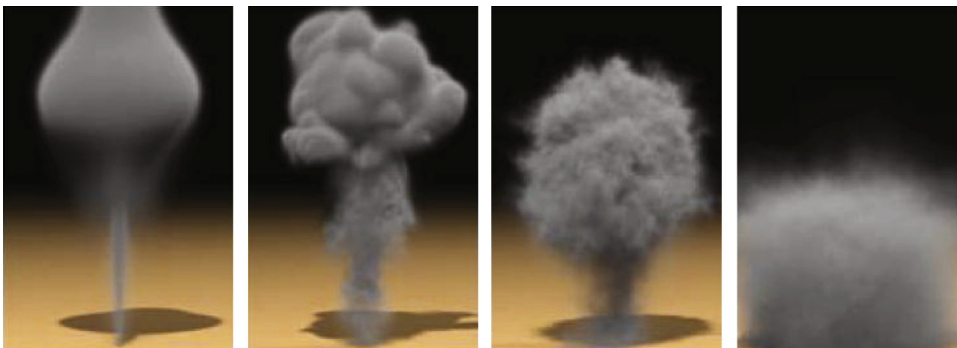
Many other methods use high-level turbulence models to compensate for sub-grid details loss due to the numerical dissipation. The basic mean flow is simulated on coarse

grid while turbulent details are synthesized using procedural synthesis methods. The methods can be described as [41]:

$$\mathbf{u} \leftarrow NS(\mathbf{U}) \oplus ST(\mathbf{u}') \quad (14)$$

where  $NS$  is a basic fluid simulator to calculate mean velocity  $\mathbf{U}$  on the coarse grid,  $ST$  is a synthesizer to generate turbulent field  $\mathbf{u}'$  and  $\oplus$  is the integration operation to couple the two fields together.

Early methods [42–45] used Kolmogorov spectrum to generate pseudo-random turbulence in frequency space. These methods have a common drawback of generating turbulence at incorrect regions. Recently curl operation [46] is applied on Perlin [9,47] and wavelet vector noises [48] to generate turbulent components at various frequencies and scales. Energy cascade is modeled to include spatial distribution and turbulence dynamics using local assembled wavelets [48], linear  $k-\varepsilon$  equation [47], one-equation [9], and complete two-equation of  $k-\varepsilon$  [49,10]. Instead of directly layering turbulence components on the basic flow, Zhao et al. [41] modeled fluctuations as controllable turbulence force to agitate the basic flow. The method guarantees temporal consistence with the basic NSEs simulators. Rather than calculating the basic flow, Gregson et al. [50] employed a multi-scale tracking method to reconstruct temporally coherent velocity field from prior low-resolution density images. However, the



**Fig. 12.** Smoke simulations with different amount of vorticity confinement forces. From left to right the coefficient is 0 (without vorticity confinement), 0.25, 0.5 and 2.0 respectively [11].

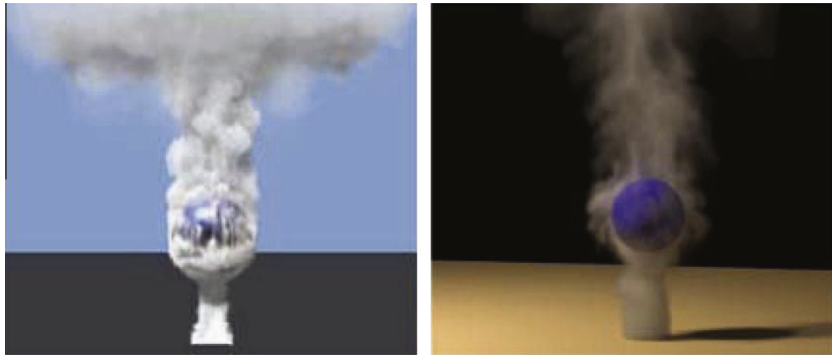


Fig. 13. Simulations using the vorticity confinement methods with constant (right) [8] and variable coefficients (left) [7].

method requires fine tuning to obtain good balance between reconstructed details and noise levels. Fig. 14 gives visual comparisons between different turbulence enhancement methods. It shows that more turbulent details are generated when more turbulence components are added to the basic flow. However, smoke appears chaotic if too much turbulence components are added. In order to obtain realistic detailed features, trial and error effort are required to determine the amount of turbulence components.

Procedural turbulence methods decouple high-frequency components from the basic flow and use synthesized small-scale turbulence to compensate for fine feature loss caused by the numerical dissipation. The methods enable animation artists to generate a fast basic simulation and then add details without changing gross motion, which helps the artists to shorten the turnaround. However, as the finest scale is determined by local velocity and temporal averaging rather than the grid resolution,

Using a much high-resolution grid is able to produce more realistic visual results than the non-physical synthesis methods. Additionally, the temporal averaging operations may smooth out visual appearance to some extent.

#### 4.4. Invariants conservation

Another baseline of reducing the numerical dissipation is to conserve basic fluid invariants. In rigid and deformable body simulations, great improvements can be achieved by following the linear and angular momenta conservation principles. Similar strategy can be applied in fluid simulation by conserving invariants including vorticity, circulation, and energy. Elcott et al. [4,51] proposed to conserve circulation along arbitrary simplicial mesh. As vortex is an important visual cue of smoke, the method uses the back-tracking integrals to solve the vorticity formulation of NSEs. The method preserves discrete circulation on grid mesh to achieve no vorticity numerical

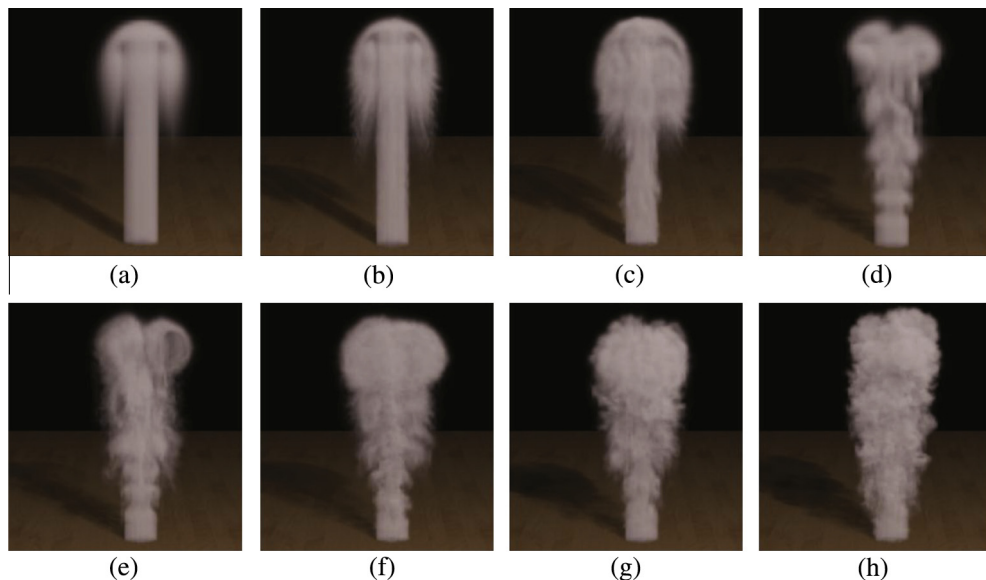


Fig. 14. Visual effects of different turbulent enhancement methods. (a) Original coarse simulation; (b) wavelet subgrid turbulence; (c) controllable and intermittent turbulence method; (d) Add vorticity confinement to (a); (e) wavelet turbulence to (d); (f)–(h) controllable and intermittent turbulence method with  $q = 0.8, 0.2$  and  $0.1$  respectively, where  $q$  is control parameter to calculate resultant velocity field with  $q\mathbf{U} + (1 - q)\mathbf{u}'$  [41].

dissipation. As shown in Fig. 15, the conservation method is able to produce similar visual results as real smoke while the basic semi-Lagrangian method loses most fine features due to the excessive numerical dissipation.

Since the method is constructed based on the semi-Lagrangian advection, it is also unconditional stability. In addition, the vorticity-based simulation requires no pressure calculation. Intrinsic storage without reference to global and local coordinate system constructs a very sparse linear system, which greatly decreases computational cost comparable with the semi-Lagrangian method. However, the method preserves the circulation but does not conserve energy. Although a  $L_2$  projection [51] can be employed to preserve the total vorticity energy, the global operation propagates influence to the entire domain. In addition, the projection induces artificial visual effects around boundaries.

In fluid simulation, the energy loss relates to time step, grid spacing, and total frame numbers. The factors are coupled in the simulation course, making it difficult to eliminate energy dissipation by tuning each single term. Besides, the computational cost is sensitive to the spatial resolution and the time step. In order to preserve energy without significantly increasing computational cost, Mullen et al. [16] proposed a fully Eulerian integration

scheme independent of spatiotemporal resolutions. The energy conservation scheme [52,53] and the non-dissipative advection ensure no energy loss on arbitrary simplicial grids. In addition, it guarantees unconditional numerical stability. Fig. 16 shows energy dissipation of different schemes over time. The energy is tightly related with the smoke motion. High energetic smoke generally obtains rich and long-time active details. As shows in the figure, the basic semi-Lagrangian method dissipates most energy over time, while the energy conservation method almost diffuses no energy, which indicates it does not have the numerical dissipation problem.

The Eulerian integration preserves energy exactly over time. However, it inevitably suffers from certain numerical diffusion due to the low-pass filtering operations of resampling on the discrete grid, which is a common drawback of grid-based methods.

#### 4.5. Particle based methods

Pure Eulerian methods start with a field sampled on grid points and end with advected field resampled on the grid at each step. The Eulerian methods may reduce numerical dissipation to some extent, but suffer from the fundamental problem of field resampling: the scale of

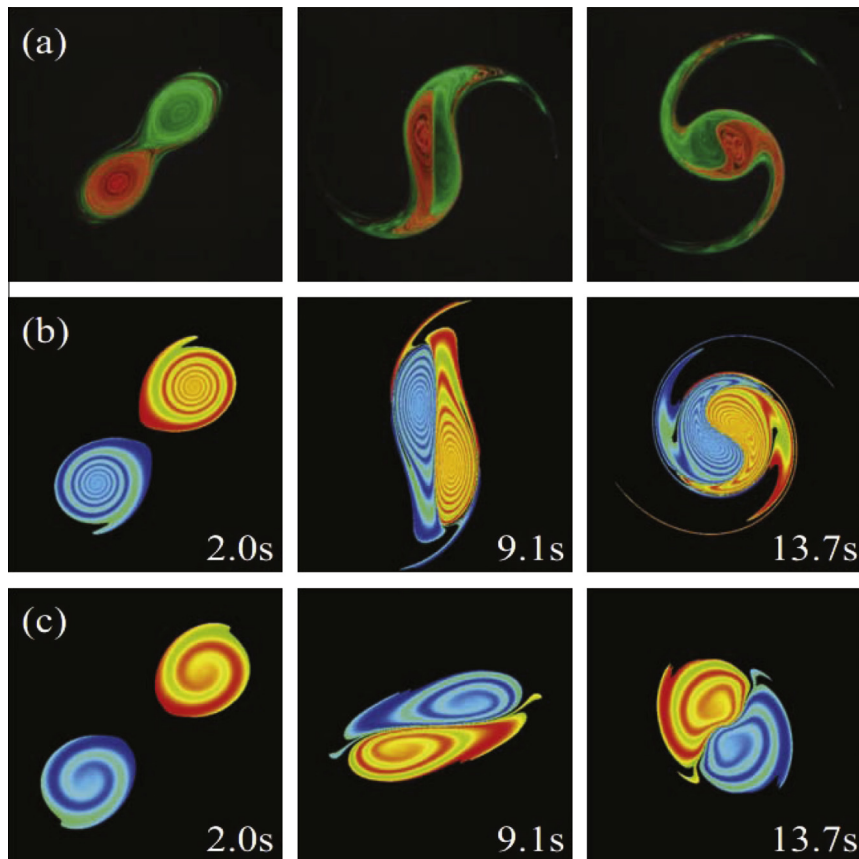


Fig. 15. 2D vortices merging comparisons between the real experimental result (a), the conservation method (b), and the basic semi-Lagrangian method (c) [4].

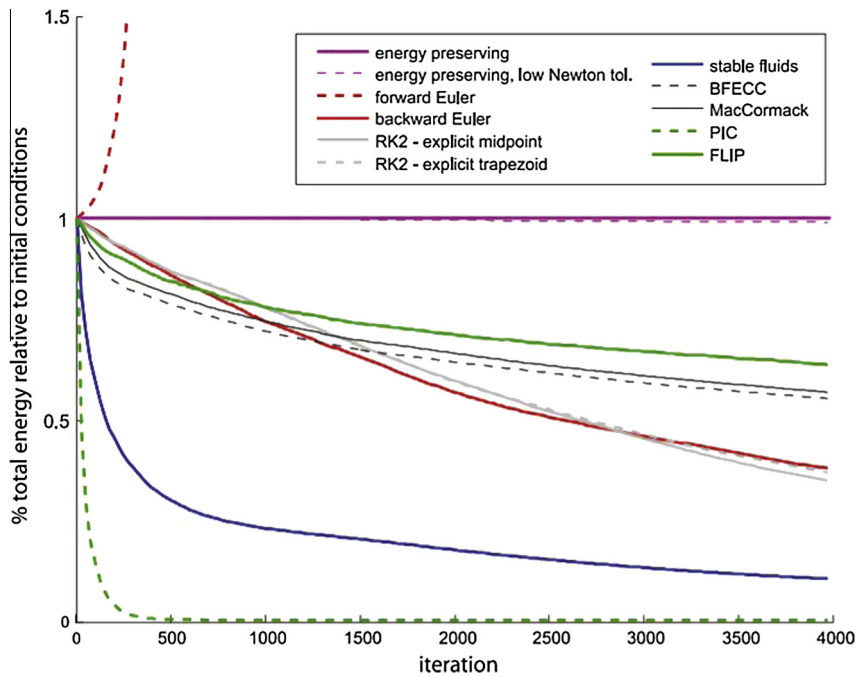


Fig. 16. Most schemes dissipate a significant amount of energy over time while the fully Eulerian scheme preserves energy exactly [16].

finest features is limited by the Nyquist frequency of the grid. Even for a pure translation velocity field, the highest spatial frequency that can be reliably advected has period of  $4\Delta x$  [1]. A good Eulerian scheme may filter out high-frequency components as a smoothing operation while a bad Eulerian scheme may further generate visual artifacts.

In contrast to the Eulerian methods, the Lagrangian methods employ discrete particles to carry variables along without dissipation. There is no filtering and dissipation loss in advection. From this point of view, Lagrangian particle methods are perfect for advection to eliminate numerical dissipation. In fact, there are already some pure particle methods for fluid simulation, such as smoothed particle hydrodynamics (SPH) [54] and some pure vortex particle methods [55]. Several vortex particle methods employ geometry structures to describe vortex structure including filament [56,57], ring [58,59] and sheet [60] to capture various subtle features such as leapfrogging vortex rings and vortex shedding, which are difficult to simulate using the grid-based methods. However, apart from their specified problems such as particle redistribution, pure particle methods cannot enforce incompressible condition as efficient and accurate as the grid methods. Besides, the particle-based methods require sophisticated particle distribution to obtain good performance and quality.

A preferable remedy is to add Lagrangian machinery to the Eulerian scheme in a hybrid way. Selle et al. [11] combined Lagrangian vortex particle and Eulerian grid together to overcome the weakness of both methods. The vortex particles are used to carry and preserve vorticity without dissipation loss while the Eulerian grid is employed to calculate velocity and vorticity confinement force [7] to

recover swirling details. As the vortex particles are only used to increase visual details, no particle distribution is required. Yoon et al. [61] proposed a similar scheme with combination of the vortex particles and Eulerian grid. Basic flow is calculated on relative coarse grid to reduce the computational cost. Particle vorticity is transferred to a high-resolution vorticity field to obtain fine details. The method uses a kernel function to calculate vorticity field on high-resolution grid points, which cannot guarantee the vorticity field free of divergence. In addition, as all vortex particles make their contributions to vorticity forces, it likely synthesizes strong vorticity forces to produce visual artifacts. The vortex particles are suitable to preserve swirling features but require additional computations to induce velocity from vorticity. A straightforward modification is to directly store velocity on particles. However, it is far from enough as pressure projection requires velocity information of the whole domain, which means we cannot just consider individual particles but account for particle-particle interactions on top of advectons. Particle-in-cell (PIC) [62] is a good strategy to couple particle and grid while solving the pressure projection problem. Velocity and other quantities are defined and advected on particles to eliminate the numerical dissipation, but they are calculated and adjusted by pressure gradients on the grid points. The method reduces the numerical dissipation of advection but introduces even more to the total numerical dissipation of the simulation because quantities are averaged to the grid points from the particles and then interpolated back to the particles from the grid points. Zhu and Bridson [12] proposed the fluid implicit particle (FLIP) method to reduce the total numerical dissipation. The FLIP interpolates the changes rather than the quantity values on the

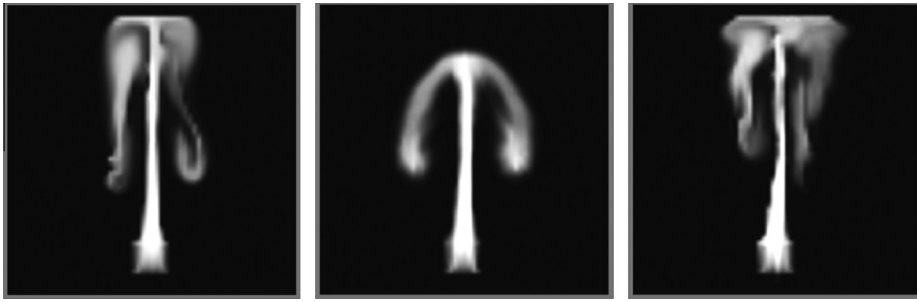


Fig. 17. Smoke simulations with semi-Lagrangian (left), PIC (middle), and FLIP (right) schemes [63].

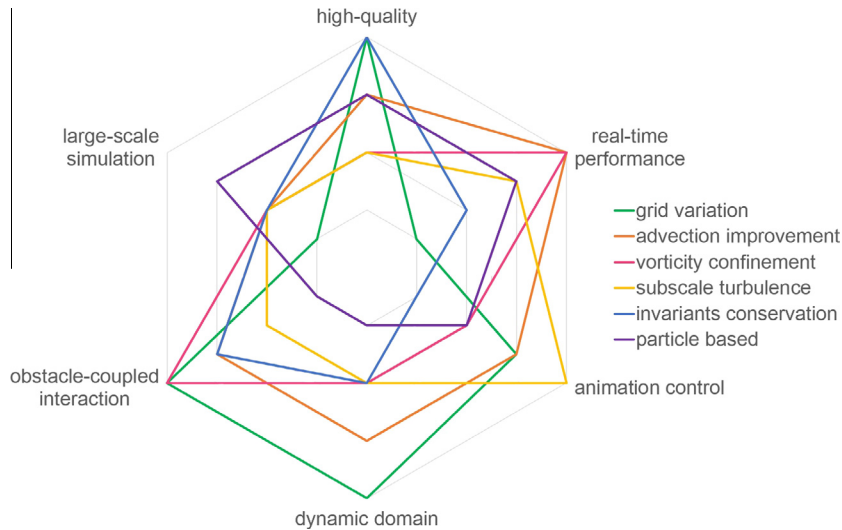


Fig. 18. The applicability of different types of methods in several typical fluid simulation scenarios. The applicability is classified into four levels with weak, moderate, suitable, and strong from inside to outside.

grid points. It adds the changes to the quantity value other than replacing it. The changes are smoothed but not accumulated to achieve almost no numerical dissipation. Fig. 17 shows the visual comparison of smoke using different methods. The FLIP method obtains much more visual details than both the basic semi-Lagrangian method and the PIC method.

As only changes are transferred in the FLIP method, the velocity fluctuations on particles may average down to zero on the grid points and then show up as unexpected perturbations at other time steps, leading to certain noise. Since PIC has no such problem, a better solution is to combine them together with a tunable regularization parameter to decay the noise and suppress the numerical dissipation.

## 5. Conclusion

In this paper, we discussed the numerical dissipation problem for smoke simulation in computer graphics community. The numerical dissipation is widely regarded to have visual consequences of visual results. Many methods have been developed to solve the problem. The methods

reduce numerical dissipation or compensate for visual loss from various aspects. Most methods fall into three categories: “Decreasing”, “Conserving”, and “Increasing”. Table 3 compares methods in the categories in terms of computational cost, visual quality, implementation, adaptability, and scalability. The adaptability indicates whether the method is adaptive to different application scenarios. The scalability measures how it is easy to integrate the method with existing applications. The following list gives the common characteristics of methods in each category, which has its advantages and disadvantages to make them suitable for different application scenarios including high-quality visual effect, real-time performance, animation control, dynamic domain, obstacle-coupled interaction, and large-scale simulation, as illustrated in Fig. 18.

- “Decreasing”: Solving the problem by reducing the amount of numerical dissipation. The grid variation methods in Section 4.1 and the advection improvement methods in Section 4.2 fall into this category. The “decreasing” methods normally use fine-resolution discretization and high-order numerical schemes to reduce the total amount of the numerical dissipation. As grid

**Table 3**

Method comparison in terms of computational cost, visual quality, implementation, adaptability, and scalability.

Category	Method	Computational cost	Visual quality	Ease to implement	Adaptability	Scalability
Decreasing	[3]	Low	Moderate	Moderate	Moderate	Moderate
	[4]	High	High	Moderate	High	Low
	[5]	Moderate	High	Moderate	High	Low
	[6]	Low	Low	Easy	Low	Low
	[17]	Low	Moderate	Easy	Moderate	Low
	[18]	Moderate	High	Moderate	High	Moderate
	[19]	High	High	Hard	High	Low
	[20]	High	High	Hard	High	Low
	[1]	Low	Low	Easy	High	High
	[7] (mono. interp.)	Moderate	Moderate	Easy	High	High
	[21]	Low	Moderate	Easy	High	High
	[24,25]	High	Moderate	Easy	High	High
	[23]	Moderate	Moderate	Easy	High	High
	[26]	Moderate	High	Easy	High	High
	[32,33]	High	Moderate	Moderate	Moderate	Moderate
	[35]	Moderate	High	Moderate	High	High
Conserving	[4,51]	High	High	Hard	Moderate	Low
	[16]	High	High	Hard	High	Low
	[12]	Moderate	High	Easy	High	High
	[11] (vort. part.)	Moderate	Moderate	Moderate	Moderate	High
	[61]	High	Moderate	Moderate	Moderate	High
Increasing	[7,11] (vort. confin.)	Moderate	Moderate	Moderate	High	High
	[8,40]	Moderate	Moderate	Moderate	High	High
	[46]	Moderate	Moderate	Easy	High	High
	[47]	Moderate	Moderate	Moderate	Moderate	Moderate
	[48]	High	Moderate	Moderate	Moderate	Moderate
	[9]	High	Moderate	Hard	Moderate	High
	[10,49]	High	High	Hard	Moderate	High
	[41]	High	High	Moderate	High	High
	[50]	High	High	Hard	Low	Low

resolution is directly related to the final visual effect, grid variation methods have strong applicability in applications with high-quality effect requirement. The grid-based methods are flexible to handle boundary and obstacle problems, making grid variation methods suitable for applications with dynamic domain and obstacle-coupled interaction. However, great efforts are required to implement spatial discretization and guarantee the numerical stability on irregular grid structures. The heavy computational cost also degrades the runtime performance, making them ill-suited for realtime applications. As computation is exponentially proportional to the domain dimensions, tremendous computations and memory storage are required for large-scale simulations. Besides, most calculations and storage strategy tightly depend on the grid structures. Changing grid structures requires great changes in the simulation, while the advection improvement methods are generally orthogonal to other subroutines in the pipeline, the modifications will not impact other subroutines, so the advection improvement methods are much easier to integrate with existing applications than grid variation methods. In addition, the advection improvement methods are compatible with the grid-based methods. They are feasible to handle the dynamical domain and obstacle boundary problems with the help of grid-based methods. Comparing with the grid variation methods, advection improvement methods improve the visual quality with relative less computational overhead. They are preferable for real-time applications that do not require very high-quality results.

- “Conserving”: Solving the problem by conserving indirect fluid quantities. The invariants conservation methods in Section 4.4 and the particle methods in Section 4.5 are two typical “conserving” methods, while they preserve quantities in different ways. The invariants conservation methods integrate indirect fluid quantities (e.g. circulation and energy) on specific structures (e.g. simplicial mesh) to avoid simulating direct quantities that greatly impact fluid motions. As direct quantities such as velocity are not directly involved in calculations that produce the numerical dissipation, the numerical dissipation is significantly reduced, making them especially suitable for applications with high-quality requirement. Especially, as no energy is dissipated for the energy conservation methods, smoke remains energetic to guarantee long-time performances. The invariants conservation methods are usually constructed on the grid structures, so they may inherit the advantages of the grid variation methods. However, they need sophisticated strategies to guarantee conservation, which impeding their adaption and stability to other application scenarios. The particle methods preserve quantity by leveraging the conservation characteristic of the particle. The particle-based methods have totally different simulation pipelines from grid-based methods, leading to different applicabilities. The particle-based methods are difficult to handle boundary problems such as obstacle interaction and dynamic domain. Hybrid particle and grid methods alleviate the problems. The particle based methods do not require some computational intensive subroutines,

which reduces the computational cost to make them suitable for realtime applications. Comparing to the grid point number of grid-based methods, particle number is more relaxant to the domain dimensions, making particle methods applicable to large-scale simulations with less computational cost.

- “Increasing”: Solving the problem by increasing additional components. The compensation methods in Section 4.3 are typical “increasing” methods as they add components back into flow to synthesize detailed features. The vorticity confinement methods are usually directly integrated into the grid methods as an additional term. They need no modification to original grid structure and do not impact original simulation. As computational cost is inexpensive, the vorticity confinement methods are able to increase diversity of flow motion with much few efforts. They have the same applicability as the grid-based methods. Most sub-grid compensation methods employ the grid-based methods for the basic flow simulations, so they have similar abilities to the grid-based methods. However, the requirement to guarantee consistence between the basic flow and the sub-grid synthesis flow degrades their abilities. Especially, computational cost increases significantly when complicated turbulence models are used on sub-grid structure to generate turbulent details. A great advantage of the “increasing” methods is that the synthesis procedure is decoupled from the basic simulation. The synthesis procedure can be treated as a postprocessing without influencing the gross motion, which makes the “increasing” methods suitable for animation control. However, As turbulent details are generally generated using synthesis methods, the details may appear artificial and less comparable with the grid-based methods.

In most time, we struggle to eliminate the numerical dissipation, while occasionally we leverage it to guarantee the numerical stability. As smoke is typically simplified as inviscid fluid, there is no viscous friction at the dissipative range of energy cascading. If there was no numerical dissipation acting as artificial viscosity, energy would pile up unbounded at the finest scales (with highest wavenumber as  $k = 2\pi/\Delta x$ ) and cause numerical instability [49]. Additional dissipations are required to maintain numerical stability. The popular semi-Lagrangian scheme is so much dissipative over a wide range of energy spectrum that it degrades the visual quality. As we have observed in the full Eulerian integration method [16], visual results may look unnatural if there is no numerical dissipation. Adding a certain amount of numerical dissipation will make it a little viscous and appear plausible as real smoke.

## References

- [1] R. Bridson, *Fluid Simulation for Computer Graphics*, AK Peters, Ltd., 2008.
- [2] J. Stam, Stable fluids, in: Proc. SIGGRAPH '99, 1999, pp. 121–128.
- [3] F. Losasso, F. Gibou, R. Fedkiw, Simulation water and smoke with an octree data structure, in: Proc. SIGGRAPH'04, vol. 23(3), 2004, pp. 457–462.
- [4] S. Elcott, Y. Tong, E. Kanso, Discrete, circulation preserving, and stable simplicial fluids, *ACM Trans. Graph.* 26 (1) (2005).
- [5] B.E. Feldman, F. James, M. Bryan, Animating gases with hybrid meshes, in: Proc. SIGGRAPH'05, vol. 24(3), 2005, pp. 904–909.
- [6] M. Shah, J. Cohen, S. Patel, P. Lee, F. Pighin, Extended galilean invariance for adaptive fluid simulation, in: Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2004, pp. 13–21.
- [7] R. Fedkiw, J. Stam, H.W. Jensen, Visual simulation of smoke, in: Proc. SIGGRAPH '01, 2001, pp. 15–22.
- [8] S. He, H.C. Wong, W.M. Pang, U.H. Wong, Realtime smoke simulation with improved turbulence by spatial adaptive vorticity confinement, *Comput. Anim. Virt. Worlds* 22 (2) (2011) 107–114.
- [9] R. Narain, J. Sewall, M. Carlson, M.C. Lin, Fast animation of turbulence using energy transport and procedural synthesis, in: Proc. SIGGRAPH Asia '08, 2008, pp. 1–8.
- [10] T. Pfaff, N. Thuerey, J. Cohen, S. Tariq, Scalable fluid simulation using anisotropic turbulence particles, *ACM Trans. Graph.* 29 (6) (2010) 174–182.
- [11] A. Selle, N. Rasmussen, R. Fedkiw, A vortex particle method for smoke, water and explosions, *ACM Trans. Graph.* 24 (3) (2005) 910–914.
- [12] Y. Zhu, R. Bridson, Animating sand as a fluid, in: Proc. SIGGRAPH '05, 2005, pp. 965–972.
- [13] M.J. Harris, Fast fluid dynamics simulation on the GPU, *GPU Gems 1* (2004) 637–665.
- [14] M. Berger, J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484–512.
- [15] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84.
- [16] P. Mullen, K. Crane, D. Pavlov, Y.Y. Tong, M. Desbrun, Energy-preserving integrators for fluid animation, in: Proc. SIGGRAPH'09, vol. 28(3), 2009.
- [17] J. Cohen, S. Tariq, S. Green, Interactive fluid particle simulation using translating eulerian grids, in: Proc. SIGGRAPH Symposium on Interactive 3D Graphics and Games, 2010.
- [18] B. Zhu, W.L. Lu, M. Cong, B. Kim, R. Fedkiw, A new grid structure for domain extension, in: Proc. SIGGRAPH'13, vol. 32(4), 2013.
- [19] B.E. Feldman, J.F. O'Brien, B.M. Klingner, T.G. Goktekin, Fluids in deforming meshes, in: Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2005, pp. 255–259.
- [20] B.M. Klingner, B.E. Feldman, N. Chentanez, J.F. O'Brien, Fluid animation with dynamic meshes, in: Proc. SIGGRAPH'06, 2006, pp. 820–825.
- [21] Z.P. Huang, L. Han, G.H. Gong, A local adaptive Catmull-Rom to reduce numerical dissipation of semi-lagrangian, *Comput. Anim. Virt. Worlds* (2013).
- [22] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, *Int. J. Numer. Methods Fluids* 72 (811–845) (2013).
- [23] A. Selle, R. Fedkiw, B. Kim, Y.Y. Liu, J. Rossignac, An unconditionally stable MacCormack method, *J. Sci. Comput.* (2008).
- [24] B. Kim, Y. Liu, I. Llamas, R. Rossignac, Flowfixer: using bfec for fluid simulation, in: Proc. Eurographics Workshop on Natural Phenomena, 2005.
- [25] B. Kim, Y. Liu, I. Llamas, Advections with significantly reduced dissipation and diffusion, *IEEE Trans. Visual. Comput. Graph.* 13 (1) (2007) 135–144.
- [26] J. Molemaker, M.C. Jonathan, P. Sanjit, J.Y. Noh, Low viscosity flow simulations for animation, in: Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2008.
- [27] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, *Comput. Meth. Appl. Mech. Eng.* 19 (1979) 59–98.
- [28] C. Canuto, M.Y. Hussaini, A. Quarteroni, T. Zang, *Introduction to Computer Graphics*, Springer-Verlag, 1988.
- [29] J.H. Ferziger, M. Peric, *Computational Methods for Fluid Dynamic*, Springer-Verlag, 1996.
- [30] T. Yabe, T. Aoki, A universal solver for hyperbolic equations by cubic-polynomial interpolation. I: One-dimensional solver, *Comput. Phys. Commun.* 66 (1991) 2–3.
- [31] T. Yabe, T. Ishikawa, P.Y. Wang, T. Aokit, Y. Kadota, F. Ikeda, A universal solver for hyperbolic equations by cubic-polynomial interpolation. II: Two and three-dimensional solvers, *Comput. Phys. Commun.* 66 (1991) 233–242.
- [32] O.Y. Song, H. Shin, H.S. Ko, Stable but non-dissipative water, *ACM Trans. Graph.* 24 (2005) 81–97.

- [33] O.Y. Song, D. Kim, H.S. Ko, Derivative particles for simulating detailed movements of fluids, *IEEE Trans. Visual. Comput. Graph.* 13 (2007) 711–719.
- [34] F.X. F. T. Yabe, G. Nizam, T. Ito, Constructing a multi-dimensional oscillation preventing scheme for the advection equation by a rational function, *Comput. Phys. Commun.* 94 (1996) 103–118.
- [35] D. Kim, S.O. Young, A semi-lagrangian CIP fluid solver without dimensional splitting, *Comput. Graph. Forum* 27 (2) (2008) 467–475.
- [36] S. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comp. Phys.* 31 (1979) 335–362.
- [37] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (1) (2002) 83–116.
- [38] Z.P. Huang, G.H. Gong, L. Han, Physically-based modeling, simulation and rendering of fire for computer animation, *Multimedia Tools Appl.* (2012).
- [39] J. Steinhoff, D. Underhill, Modification of the euler equations for vorticity confinement: application to the computation of interacting vortex rings, *Phys. Fluids* 6 (8) (1994) 2738–2744.
- [40] S. He, R. Lau, Synthetic controllable turbulence using robust second vorticity confinement, *Comput. Graph. Forum* 32 (1) (2013) 27–35.
- [41] Y. Zhao, Z. Yuan, F. Chen, Enhancing fluid animation with adaptive, controllable and intermittent turbulence, in: *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, pp. 75–84.
- [42] J. Stam, E. Fiume, Turbulent wind fields for gaseous phenomena, in: *Proc. SIGGRAPH '93*, 1993, pp. 369–376.
- [43] J. Stam, A General Animation Framework for Gaseous Phenomena, ERCIM Research Report, 1997.
- [44] A. Lamorlette, N. Foster, Structural modeling of flames for a production environment, *ACM Trans. Graph.* 21 (3) (2002) 729–735.
- [45] N. Rasmussen, D.Q. Nguyen, Geigerw, R. Fedkiw, Smoke simulation for large scale phenomena, *ACM Trans. Graph.* 22 (3) (2003) 703–707.
- [46] R. Bridson, J. Hourihan, Nordenstamm, Curl noise for procedural fluid flow, in: *Proc. SIGGRAPH '07*, 2007, p. 46.
- [47] H. Schechter, R. Bridson, Evolving sub-grid turbulence for smoke animation, in: *Proc. Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2008, pp. 1–8.
- [48] T. Kim, N. Thurey, D. James, M. Gross, Wavelet turbulence for fluid simulation, in: *Proc. SIGGRAPH '08*, 2008, pp. 1–6.
- [49] T. Pfaff, N. Thurey, A. Selle, M. Gross, Synthetic turbulence using artificial boundary layers, in: *Proc. SIGGRAPH Asia '08*, 2009, pp. 1–10.
- [50] J. Gregson, I. Ihrke, N. Thurey, W. Heidrich, From capture to simulation - connecting forward and inverse problems in fluids, in: *Proceedings ACM SIGGRAPH 2014*, 2014, pp. 139–150.
- [51] S. Elcott, Y. Tong, E. Kanso, P. Schroder, M. Desbrun, Stable, circulation-preserving, simplicial fluids, *ACM Trans. Graph.* 26 (5) (2007).
- [52] X. Zhang, D. Schmidt, B. Perot, Accuracy and conservation properties of a 3d unstructured staggered mesh scheme for fluid dynamics, *J. Comput. Phys.* 197 (2) (2002) 764–791.
- [53] K. Mahesh, G. Constantinescu, P. Moin, A numerical method for large-eddy simulation in complex geometries, *J. Comput. Phys.* 197 (1) (2004).
- [54] M. Muller, D. Charypar, M. Gross, Particle-based fluid simulation for interactive applications, in: *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, pp. 154–159.
- [55] S.I. Park, M.J. Kim, Vortex fluid for gaseous phenomena, in: *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 261–270.
- [56] A. Angelidis, F. Neyret, Simulation of smoke based on vortex filament primitives, in: *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 87–96.
- [57] S. Weibann, U. Pinkall, Real-time interactive simulation of smoke using discrete integrable vortex filaments, in: *Proc. Vir. Real., Inter. and Phys. Sim.*, 2009, pp. 1–10.
- [58] A. Angelidis, F. Neyret, A controllable, fast and stable basis for vortex based smoke simulation, in: *Proceedings of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, 2006.
- [59] S. Weibmann, U. Pinkall, P. Schroder, Smoke rings from smoke, *ACM Trans. Graph.* 33 (4) (2014).
- [60] T. Pfaff, N. Thurey, M. Gross, Lagrangian vortex sheets for animating fluids, *ACM Trans. Graph.* (2012) 1–8.
- [61] J.C. Yoon, H.R. Kam, J.M. Hong, S.J. Kang, C.H. Kim, Procedural synthesis using vortex particle method for fluid simulation, *Comput. Graph. Forum* 28 (7) (2009) 1853–1859.
- [62] F.H. Harlow, Procedural synthesis using vortex particle method for fluid simulation, in: *The Particle-in-Cell Method for Numerical Solution of Problems in Fluid Dynamics*, 1963, pp. 269.
- [63] C. Camero, A Comparison of Grid Based Techniques for Navier Stokes Fluid Simulation in Computer Graphics, Master thesis, University of California, San Diego, 2008.