

Dynamic BFECC Characteristic Mapping method for fluid simulations

Xiaosheng Li · Le Liu · Wen Wu · Xuehui Liu · Enhua Wu

Published online: 5 May 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this paper, we present a new numerical method for advection in fluid simulation. The method is built on the Characteristic Mapping method. Advection is solved via grid mapping function. The mapping function is maintained with higher order accuracy BFECC method and dynamically reset to identity mapping whenever an error criterion is met. Dealing with mapping function in such a way results in a more accurate mapping function and more details can be captured easily with this mapping function. Our error criterion also allows one to control the level of details of fluid simulation by simply adjusting one parameter. Details of implementation of our method are discussed and we present several techniques for improving its efficiency. Both quantitative and visual experiments were performed to test our method. The results show that our method brings significant improvement in accuracy and is efficient in capturing fluid details.

Keywords Fluid simulation · Advection · BFECC · Characteristic Mapping

1 Introduction

Fluid phenomena such as water, smoke and explosions are common in daily life. The motion of fluid is complex and full of details, which attracts increasing attention in com-

puter graphics. To simulate fluid motion realistically, many techniques were developed to solve the Navier–Stokes equations, which describe the physics of fluids.

One critical step of Navier–Stokes equations is the advection, which deals with the transportation of physical variables along the velocity field. Semi-Lagrangian method [15] was popularly used for advection as it is unconditionally stable. However, the first-order semi-Lagrangian is of low accuracy, leading to a significant amount of numerical diffusion and dissipation.

Many techniques were proposed to overcome this problem with higher order accuracy, such as vorticity confinement [5], particle level set method [3], back and forth error compensation and correction (BFECC) [2] and MacCormack method [13]. Higher order methods are normally more difficult to implement, particularly for non-uniform meshes and are more time consuming than first-order methods.

Tessendorf and Pelfrey examined the method of characteristics and presented a mathematical framework called Characteristic Mapping (CM) method for solving the advection in VFX production [16]. The advection problem was then solved via first-order semi-Lagrangian advection of mapping function. CM method reduces the numerical diffusion and dissipation and improves the accuracy of advection effectively. However, first-order semi-Lagrangian advection leads to numerical diffusion for mapping functions still. In addition, when the mapping function is badly distorted under extreme velocity field, the accuracy decreases severely. We can simply reset the mapping to identity mapping frequently. However, it is difficult to determine the frequency as excessive remapping introduces extra diffusion and dissipation while deficient remapping results in artifacts. This would be even worse when we use dynamic CFL timestep in fluid simulation. Thus we need a high-order accuracy advection for mapping function and a careful remap-

X. Li (✉) · L. Liu · X. Liu · E. Wu
State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing, China
e-mail: lixs@ios.ac.cn

X. Li · L. Liu
University of Chinese Academy of Sciences, Beijing, China

W. Wu · E. Wu
University of Macau, Macao, China

ping scheme to maintain the accuracy of the mapping function.

In this paper, instead of purely advecting the mapping function with first-order semi-Lagrangian method, we advect the mapping function with high-order BFECC method and we provide an error criterion for checking the quality of the mapping function. The mapping function is dynamically maintained and remapped. Whenever the error criterion is met, we reset the mapping function to identity mapping for better accuracy. This treatment brings significant improvement in accuracy and our error criterion allows one to control the level of details of fluid by simply adjusting one parameter.

In summary, our contribution is a dynamic BFECC Characteristic Mapping method, which points out that by maintaining the mapping function with higher order accuracy scheme and careful remapping scheme, the accuracy of Characteristic Mapping method can be significantly improved.

In the next section, we briefly discuss some related works. Section 3 gives the mathematical formulation and Sect. 4 details the numerical implementation of our method. We demonstrate results of our work in Sect. 5 and discuss about the limitations and future work in Sect. 6.

2 Related work

In fluid simulation, the stability problem of advection under Eulerian grid was successfully addressed by semi-Lagrangian advection [15], which enables the widespread applications of fluid simulation in visual effects. However, semi-Lagrangian advection suffers from built-in numerical diffusion and dissipation, i.e., small details dissipate quickly due to linear interpolation in the first-order semi-Lagrangian.

Researchers had proposed many techniques to solve this problem. Fedkiw et al. [5] presented a vortex confinement method by adding the vorticity back to the velocity field in. They also improved the accuracy by replacing the linear interpolation with a monotonic cubic interpolation. Kim et al. [9] proposed a new constrained interpolation profile (CIP) solver, which was stable and accurate. Partial derivatives should be advected to build the sub-cell profile, thus increasing the implementation complexity and computation time. Kim et al. [7, 8] introduced the BFECC method to computer graphics, which had been analyzed by Selle et al. [13]

Several hybrid approaches were proposed to combine the Eulerian and Lagrangian frameworks to overcome the fundamental drawback of grid-based interpolation. Enright et al. [3] developed a particle level set method. Selle et al. [14] used vortex particle to transport vortices without loss. Zhu and Bridson [18] introduced the FLIP method to graphics community and advection was performed with massless particles.

Hachisuka [6] presented a combined Lagrangian–Eulerian approach for accurate advection, which solved the advection using two mapping functions. Similarly, Tessendorf and Pelfrey examined the semi-Lagrangian method and formulated the mathematical framework of CM for solving the advection in VFX production [16]. The advection problem was then solved via the semi-Lagrangian advection of mapping function. CM method can reduce the numerical diffusion and dissipation and improve the accuracy of advection but suffers from accuracy loss when mapping function is distorted badly.

The mapping function was advected using first-order semi-Lagrangian advection in [16]. Contrarily, we extend the CM method by advecting mapping function with high-order accuracy BFECC method and we also present an error criterion for dynamically maintaining the mapping function with accuracy. Be noted that, [16] did mention that the advection of the mapping function can be solved by other advection schemes, however, no results were presented and they did not address the problem of remapping either. To the best of our knowledge, we are the first to apply CM to water simulation, which requires careful remapping otherwise visual artifacts would be quite obvious. We refer to our method as the dynamic BFECC characteristic mapping (DBCM) method.

Recently Mercier and Nave [10] presented a similar framework for advecting arbitrary sets in a vector field. They advected the mapping function with gradient-augmented level set (GALS) method [11], which is more complicated than BFECC and they used dynamic grid resolution for mapping function while we only use fixed resolution for the tradeoff between accuracy and computational cost. We also provide an importance sampling technique for remapping. Thus our method is much simpler to implement and efficient.

3 Mathematical formulation

In this section, we briefly introduce the theoretical foundations of Characteristic Mapping described in [10, 16] and present the mathematical formulation of the dynamic BFECC Characteristic Mapping method.

3.1 BFECC Characteristic Mapping method

Given a regular Cartesian grid, the advection is phrased as mapping function $\chi(\mathbf{x}, t)$ through a velocity field $\mathbf{u}(\mathbf{x}, t)$. $\chi(\mathbf{x}, t)$ defines the mapping from point \mathbf{x} in space at time t to its position at the initial time. $\chi(\mathbf{x}, t)$ is advected by a velocity field $\mathbf{u}(\mathbf{x}, t)$, thus the evolution of $\chi(\mathbf{x}, t)$ is formally defined as the solution of the advection problem

$$\frac{\partial \chi(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \chi(\mathbf{x}, t) = 0 \tag{1}$$

$$\chi(\mathbf{x}, 0) = \mathbf{x} \tag{2}$$

Equation (2) defines the initial condition of the mapping function, which is the identity mapping. All the information necessary for advecting a set is contained in this map. For any initial set function S_0 , e.g., level set functions $S_0(\mathbf{x}) = \phi_0(\mathbf{x})$, evolved under velocity field \mathbf{u} , the set at a final time T can be written as

$$S(\mathbf{x}, T) = S_0(\chi(\mathbf{x}, T)) \tag{3}$$

Equation (3) is not restricted to level set function. In general, any set functions can be advected via this mapping procedure, which makes the CM method highly efficient, parallelizable and easy to implement.

To solve Eq. (1) numerically, we employ the semi-Lagrangian method. In paper [16], linear semi-Lagrangian method was used, namely,

$$\chi(\mathbf{x}, t + \Delta t) = \chi(\mathbf{x} - \mathbf{u}\Delta t, t) \tag{4}$$

As stated above, linear semi-Lagrangian method inherently suffers from severe numerical diffusion and dissipation. Therefore the mapping function is of low accuracy in such case.

To remedy this problem, we advect the mapping function with BFECC method. BFECC advects the solution forward and then backward in time and compares the result to the original data to estimate the error. The error estimate is then used to correct the data before advection to raise the accuracy to second order.

Let A be the forward operator used in Eq. (4) from time t to time $t + \Delta t$, and A^R be the backward operator, we can get equations below

$$\chi(\mathbf{x}, t + \Delta t) = \chi(\mathbf{x} - \mathbf{u}\Delta t, t) = A(\chi(\mathbf{x}, t)) \tag{5}$$

$$\chi(\mathbf{x}, t) = \chi(\mathbf{x} + \mathbf{u}\Delta t, t + \Delta t) = A^R(\chi(\mathbf{x}, t + \Delta t)) \tag{6}$$

BFECC advection for Eq. (1) can be formulated as follows:

Step 1. Solve $\hat{\chi}(\mathbf{x}, t + \Delta t)$ using $\hat{\chi}(\mathbf{x}, t + \Delta t) = A(\chi(\mathbf{x}, t))$.

Step 2. Solve $\hat{\chi}(\mathbf{x}, t)$ using $\hat{\chi}(\mathbf{x}, t) = A^R(\hat{\chi}(\mathbf{x}, t + \Delta t))$.

Step 3. Let $\bar{\chi}(\mathbf{x}, t) = \chi(\mathbf{x}, t) + (\chi(\mathbf{x}, t) - \hat{\chi}(\mathbf{x}, t))/2$

Step 4. Solve $\chi(\mathbf{x}, t + \Delta t)$ using $\chi(\mathbf{x}, t + \Delta t) = A(\bar{\chi}(\mathbf{x}, t))$.

BFECC method can be implemented very easily and exhibits second-order accuracy in both space and time [8]. With BFECC, the accuracy of mapping function can be significantly improved thus resulting in a more accurate Characteristic Mapping method.

3.2 Dynamic remapping

Theoretically, we can evolve the mapping function forward to any time. However, the mapping function may be distorted severely by the velocity field \mathbf{u} , which increases the interpolation error. Thus, we need to reset the mapping function to identity mapping (i.e., remapping) sometime after the evolution starts.

When to remap is not trivial as the distortion depends on specific characteristics of the velocity field \mathbf{u} . Remapping frequently, such as every 10 steps, is a simple solution. However, this solution cannot achieve the desired results as it is difficult to decide the frequency well. Excessive remapping introduces extra diffusion and dissipation while deficient remapping causes artifacts. In both cases, the mapping function suffers from severe accuracy loss.

To control the representation error induced by the mapping function, we want to be able to detect the situation where the interpolation error of χ becomes larger than a predefined tolerance (or threshold) ϵ .

To evaluate this error, we first present another mapping function $\chi^R(\mathbf{x}, t)$, which defines the mapping from point \mathbf{x} at initial time to its advected position at time t . Note that $\chi(\mathbf{x}, t)$ defines the mapping from time t to initial time while $\chi^R(\mathbf{x}, t)$ defines the mapping from initial time to time t reversely.

Unlike χ , we evolve χ^R directly by solving the set of ODEs

$$\frac{\partial \chi^R(\mathbf{x}, t)}{\partial t} = \mathbf{u}(\mathbf{x}, t) \tag{7}$$

$$\chi^R(\mathbf{x}, 0) = \mathbf{x} \tag{8}$$

χ^R is evolved at the same time when χ evolved with the same initial condition. We solve Eq. (7) by using a sufficiently accurate four-order Runge–Kutta solver (RK4), which can be regarded as advecting Lagrangian particles located in grid cells.

Augmented with $\chi^R(\mathbf{x}, t)$, we are now ready to measure the interpolation error of $\chi(\mathbf{x}, t)$ with

$$M(\chi(\mathbf{x}, t)) := \max_{\mathbf{x}} \|\chi(\chi^R(\mathbf{x}, t), t) - \mathbf{x}\| \tag{9}$$

Equation (9) evaluates the error in point \mathbf{x} by first evaluating its advected position with χ^R and then evaluating the initial position of the advected position. The initial position is finally compared with \mathbf{x} to measure the error. This method makes use of the information brought by the Lagrangian treatment of χ^R .

Equipped with the error measure M , we can evaluate the quality of the χ easily. When χ induces a representation error greater than ϵ , that is

$$M(\chi(\mathbf{x}, t)) > \epsilon \tag{10}$$

We stop evolving this mapping and reset χ and χ^R to identity mapping. For any set S that is under evolution, we also reset the initial set to current set, namely $S_0(\mathbf{x}, 0) = S(\mathbf{x}, t)$. After remapping, the time t is set to zero and the evolution continues with the new mapping.

Note that we use similar techniques as [10] for dynamic remapping. But they evolve another set of particles while we use another grid, which enables the importance sampling technique described in Sect. 4.2.

In the next section, we will discuss in more details about the practical implementation of the dynamic BFECC Characteristic Mapping method.

4 Implementation

4.1 Algorithm

We first summarize our method in pseudo-code for ease of implementation in Algorithm 1. In addition, we discuss some practical issues for convenience and efficiency of the implementation.

Given a problem of advection, we first define $\chi(\mathbf{x})$ and $\chi^R(\mathbf{x})$ on two Eulerian grids (with resolution of N_c and N_f) and initialize both to identity mapping. Note that in a practical implementation, the resolution of $\chi(\mathbf{x})$ and $\chi^R(\mathbf{x})$ can be decoupled. We have found that the grid of $\chi(\mathbf{x})$ can be quite coarse while that of $\chi^R(\mathbf{x})$ should be finer (typically, $N_f = 2N_c$ or $N_f = 4N_c$). $\chi(\mathbf{x})$ and $\chi^R(\mathbf{x})$ are then evolved separately. The solution to the problem of advection is formulated as the mapping from initial set function to current set function.

The implementation of advection of $\chi(\mathbf{x})$ and $\chi^R(\mathbf{x})$ is the same as the ordinary routines in [7] and we use extrema limiter technique [13] to clamp local extrema to ensure the stability of BFECC. For back tracing computation in Eq. 4, we have found that it is better to use high-order Runge–Kutta solver (RK2 for this paper).

To save computation cost, linear interpolation is used through the whole process as we have already employed higher order accurate method for advection. Higher order interpolation does not improve the result a lot but brings extra expenses.

4.2 Error estimation

The key to remapping is an accurate error estimation for Eq. (9). If the grid is fine enough, we can iterate over all grid points to evaluate the maximum error. However, this is not practical and time consuming. To evaluate the maximum error induced by χ practically, we propose to estimate Eq. (9) with random sampling and importance sampling.

Algorithm 1 The DBCM method

Define $\chi(\mathbf{x}) = \mathbf{x}$ on a N_c grid and $\chi^R(\mathbf{x}) = \mathbf{x}$ on a N_f grid, $t = 0$, S_0 is given.

while Evolving **do**

 Advance simulation time with $t = t + \Delta t$.

 Advect $\chi(\mathbf{x})$ using BFECC method (Eq. (1)).

 Advect $\chi^R(\mathbf{x})$ using RK4 (Eq. (7)).

 Map S from S_0 (Eq. (3)).

 Estimate error criterion $M(\chi(\mathbf{x}))$ (Eq. (9)).

if $M(\chi(\mathbf{x})) > \epsilon$ **then**

$\chi(\mathbf{x}) \leftarrow \mathbf{x}$

$\chi^R(\mathbf{x}) \leftarrow \mathbf{x}$

$S_0 \leftarrow S$

$t \leftarrow 0$

end if

end while

For each grid cell of χ , we randomly sample N_s points and evaluate their errors. N_s is defined by the differential factor of N_c and N_f with $N_s = (N_f/N_c)^2$.

This process is further improved with importance sampling by taking the visual importance into consideration. To perform importance sampling, we define a spatially varying sizing function $\text{Size}(\mathbf{x})$. We evaluate the sizing function at grid cell of χ and randomly sample N points on χ with the sizing function. We then evaluate the errors at these sampling points and use the maximum of these errors as the estimated error of Eq. (9). N is determined by $N = nN_s$, and n is the number of cells with non-zero sizing function value.

For examples of level set water simulation in this paper, we perform error estimation in a narrow band of the interface with sizing function

$$\text{Size}(\mathbf{x}) = \max(0, 5 - |\phi(\mathbf{x})|) \quad (11)$$

where $\phi(x)$ is the level set function, 5 is the width of the narrow band.

For smoke simulation, we are only interested in where the concentration is not zero. Therefore, we use the size function as the combination of density, fuel and temperature:

$$\text{Size}(\mathbf{x}) = \max(\rho(\mathbf{x}), k_1 \cdot \text{fuel}(\mathbf{x}), k_2 \cdot \text{temp}(\mathbf{x})) \quad (12)$$

where $\rho(\mathbf{x})$ is the density function, $\text{fuel}(\mathbf{x})$ is the fuel function and $\text{temp}(\mathbf{x})$ is the temperature function. k_1, k_2 are parameters to adjust fuel and temperature to a similar scale as density.

Our method is versatile to cope with any sizing functions. Other criteria can also be imposed as sizing functions for more efficient computation.

Importance sampling makes the process of error estimation much more efficient by restricting the computation to areas that are visually important. This may decrease the accuracy of the mapping function but the result is visually accept-

able as long as we design the sizing function for visual importance carefully.

4.3 Mapping of set function

For any initial set S_0 under evolution, the set S at any time can be computed via Eq. (3). For example, sets defined on grid, such as level set, linear interpolation can be used to evaluate the right-hand side of Eq. (3) easily.

Furthermore, this decoupling of sets and mapping function allows one to advect multiple sets and any kind of sets with given velocity field \mathbf{u} at the same time.

Here we provide a variation of semi-Lagrangian contouring (SLC) [1] by coupling it to our method to capture thin sheets of water spray. With the combination, detailed thin sheets can be captured even in low-resolution water simulation.

Let S_0 be the initial water surface mesh. We evaluate $S(\mathbf{x})$ by first computing the mapping position \mathbf{x}_i with $\mathbf{x}_i = \chi(\mathbf{x})$. $S_0(\mathbf{x}_i)$ is then evaluated under the framework of SLC by directly computing the distance from \mathbf{x}_i to the initial surface mesh S_0 . \mathbf{x} is dynamically sampling and $S(\mathbf{x})$ is maintained on a distance tree. Surface mesh is finally abstracted on the tree for visualization.

In fact, the evolution of any kind of sets under the advection of velocity field \mathbf{u} can be incorporated with our method with very small modifications.

4.4 Choice of ϵ

The choice of ϵ affects the accuracy of the mapping function. Theoretically, the smaller ϵ is, the more accurate the mapping function will be, assuming the grid is fine enough. However, if the grid which χ lives on is coarse, too smaller tolerance would result in too frequent remapping instead. In our tests, we have found that setting ϵ to a range of $[\Delta x, N_c/10 \cdot \Delta x]$ is a good choice, where Δx is cell size of the grid of χ .

One nice feature of ϵ is that it provides a simple way to control the level of details. The value of ϵ is directly related to the small details that can be captured by our method. For example, in the water simulation, smaller ϵ leads to more splashing and turbulent water spray. However, sometimes we may prefer some smoother results. Our remapping strategy thus provides a simple way to achieve such a goal by adjusting ϵ (see Fig. 1).

5 Results

We have carried out several tests to demonstrate the effectiveness of our method. Simulations reported in this section were performed on a computer with 2.7 GHz CPU, 12 GB memory and a Nvidia GeForce GT 650 M graphic card. For all simulations, ϵ was set to $3\Delta x$ if not specified and the CFL number was 0.5. Rendering of smoke was performed with GPU ray-marching procedure while the rendering of water was performed with PBRT [12].

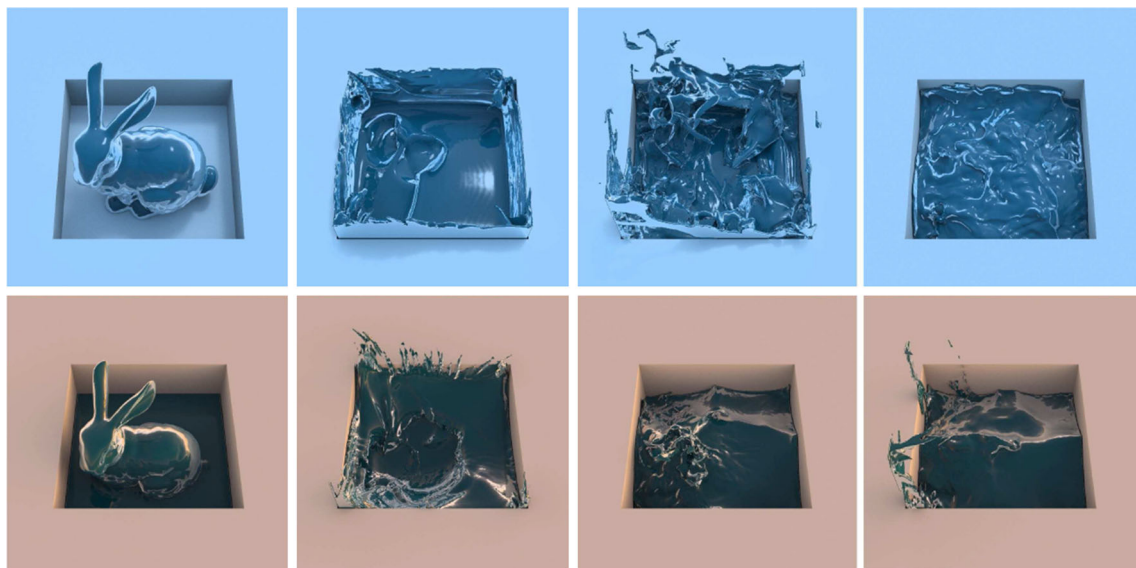


Fig. 1 Comparison of different values of ϵ . From left to right, the first one is semi-Lagrangian advection while the others used ϵ of Δx , $3\Delta x$ and $5\Delta x$ for DBCM method. Smaller ϵ led to more detailed and turbulent surface. ϵ can be used for controlling the details. As level set was

used for identifying the fluid domain, the underlying simulations could be slightly different with different values of ϵ . Even with high tolerance value, the result was still much better than that of semi-Lagrangian advection

Fig. 2 Results of Zalesak's disk (top) and Enright's sphere (bottom) after one full cycle of rotation and deformation under different advection schemes (start from the second column): 1. Semi-Lagrangian advection; 2. BFECC advection; 3. Linear CM method; 4. DBCM method. The leftmost column is the initial contour. All resolutions used were 128^3

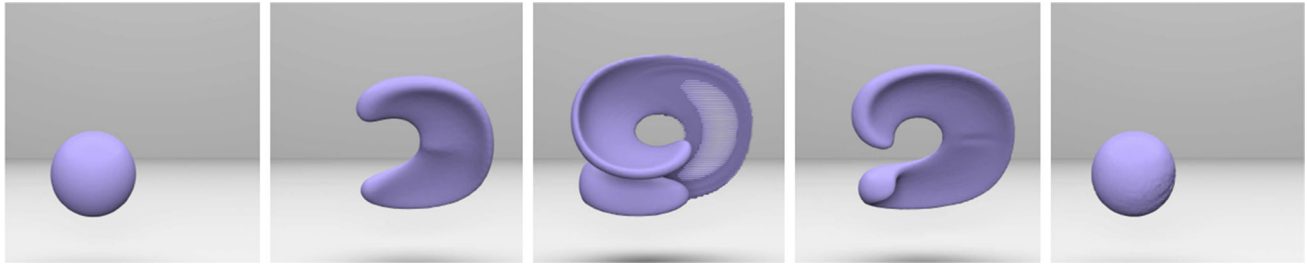


Fig. 3 Enright test of our method with a resolution of 128^3 . Note that in the extreme deformation, some details were lost and holes appeared due to the limitation of the resolution

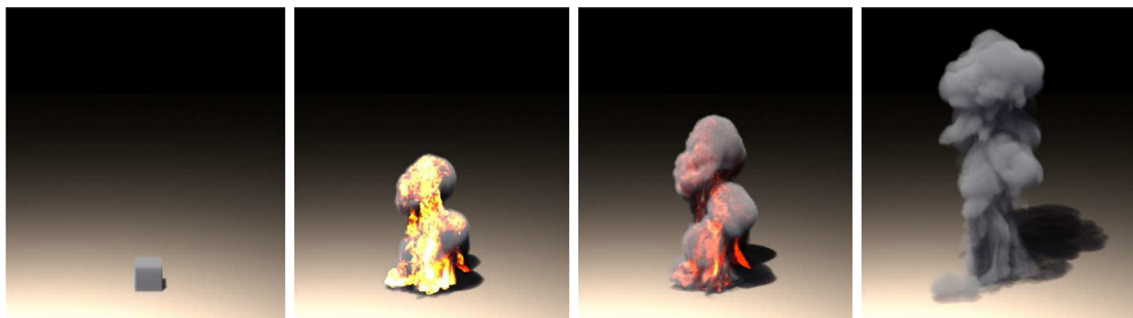


Fig. 4 Two simple 2D smoke simulations using semi-Lagrangian advection, semi-Lagrangian CM method and our method (from left to right) on a 256^2 grid. Using BFECC advection for mapping function captures the most details and the result is less diffusive than the other two methods

5.1 Basic tests

We performed two basic tests for testing the accuracy of our method for advection. We performed Zalesak's disk experiment [17] and Enright's test [3] on $128 \times 128 \times 128$ resolution grids. The contour of the disk and sphere was tracked via level set.

We used the following advection methods for these basic tests: linear semi-Lagrangian, BFECC, linear CM method and our method. Figure 2 shows the initial contours and the solutions after a full cycle of rotation and deformation. BFECC advection is effective at conserving the volume but slightly distorts the shape. Linear CM method is better at conserving the shape than BFECC but still suffers from more volume loss than BFECC. Significant improvements can be achieved by our method for conserving both shape and vol-

ume. Our method is less diffusive than the others and produces the most accurate result in these tests.

Figure 3 shows several results of the enright test using our method at different intermediate timesteps. Our method performed well in tracking the deformation of the sphere.

5.2 Smoke simulation

We applied our method to a linear semi-Lagrangian smoke simulator [15]. Our method was used to advect scalar field, such as density, fuel and temperature.

Figure 4 shows two simple 2D smoke simulations performed on a grid of 256×256 . Semi-Lagrangian advection, CM method and our method were used for these simulations. The results demonstrate that our method produced more details than the other two methods.

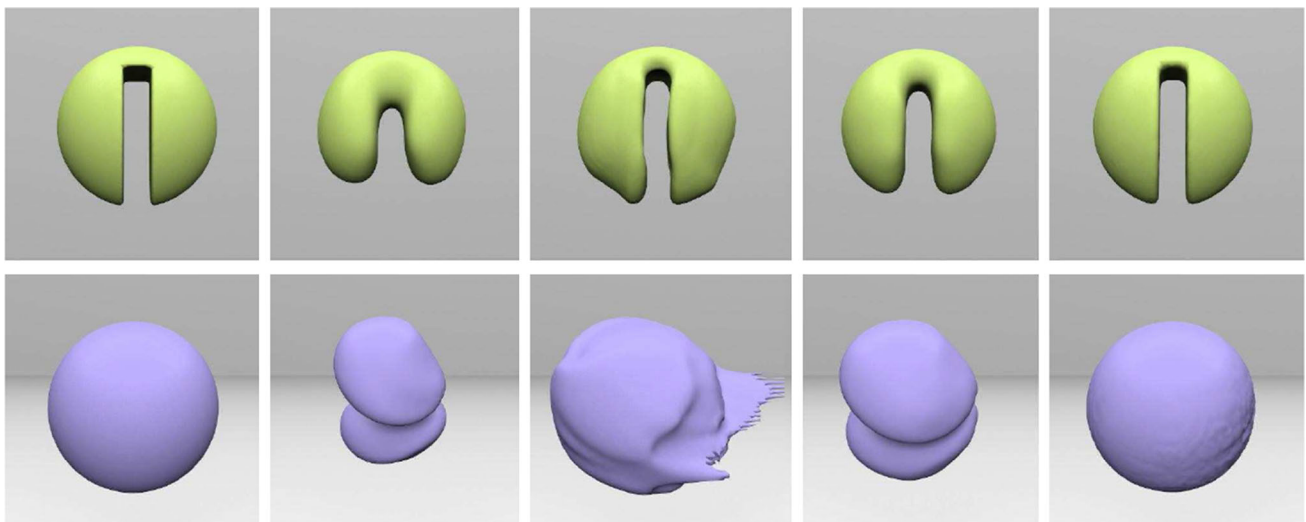


Fig. 5 Snapshots of 3D smoke simulation with linear semi-Lagrangian advection, linear CM method, DBCM method. DBCM produced a result with the most details and details of thin smoke sheets were clearly cap-

ured. Resolution of base fluid simulation was $128 \times 256 \times 128$. N_c was set to 256 for χ and N_f was set to 256 for χ^R for DBCM method



Fig. 6 Smoke simulation with fire simulation using DBCM method. Fuel and temperature field were used to simulate the fire. DBCM advected the density, fuel and temperature field simultaneously with the same mapping function and captured rich details of the distribution of these physical variables (visualized as smoke and fire). N_c was set to 256 for χ and N_f was set to 256 for χ^R for DBCM method

A 3D smoke simulation was performed similarly on a grid of $128 \times 256 \times 128$. For our method, we use resolution of $128 \times 256 \times 128$ for χ and χ^R , N_c and N_f is set to the maximum dimension, i.e., 256, for importance sampling in error estimation. The result in Fig. 5 shows that our method captured many interesting details of the smoke plume clearly.

We additionally augmented the 3D smoke simulation with fire simulation. Fire simulation was performed by adding additional variables, fuel and temperature specifically, into simulation and simulating the fuel combustion. We advected density, fuel and temperature simultaneously with our method. The result in Fig. 6 shows that we can captured

rich details of the distribution of these physical variables, which are visualized as smoke and fire.

5.3 Water simulation

For water simulation, we first used the level set water simulator in [4] and performed level set advectations with our method using χ on a grid of $128 \times 128 \times 128$ and χ^R on a grid of $256 \times 256 \times 256$. The resolution of base fluid simulation is the same as that of χ .

Figure 7 shows the results of our method applied to semi-Lagrangian level set water simulation. Our method is capable of capturing rich small details and creating turbulent water behaviors. Figure 8 shows the comparison of our method with linear semi-Lagrangian advection.

Figure 1 shows the effect of ϵ , smaller ϵ brings more accurate result and thus leads to more detailed and turbulent surface. Details can be easily controlled by adjusting the value of ϵ .

We also coupled our method to SLC method as stated in Sect. 4.3. We used base fluid simulation with a low resolution of $64 \times 64 \times 64$. Even in such low resolution, we were able to capture thin sheets of water spray with our method. We simulated two dam breaking tests to demonstrate this capability. Figure 9 shows the result of a single column of water released to collide with the walls. Thin sheets were developed when water collided with the walls. Interesting wave front was also captured in this scene. Figure 10 shows the result of two columns of water released to collide with each other and developed tall water sheets.

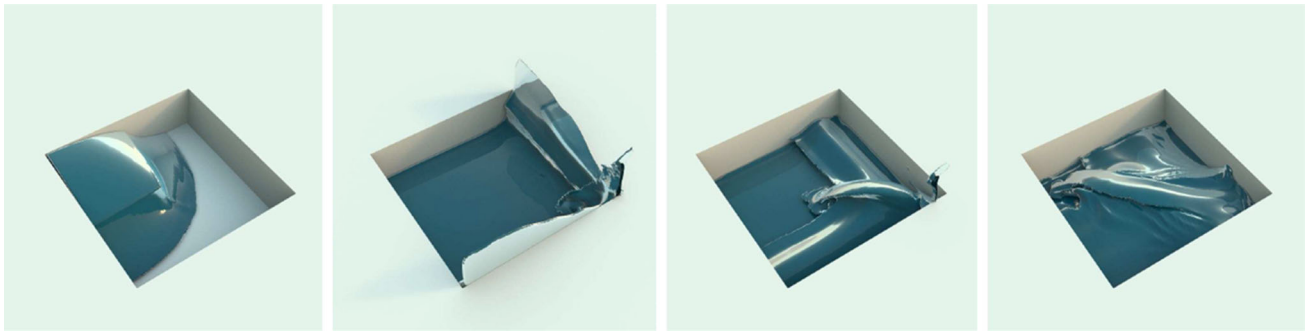


Fig. 7 Semi-Lagrangian water simulation using our method for advection of the level set. Our method is capable of capturing small details of the water. The resolution of base fluid simulation was 128^3 . N_c was

128 and N_f was 256. The bunny model is courtesy of the Stanford 3D Scanning Repository

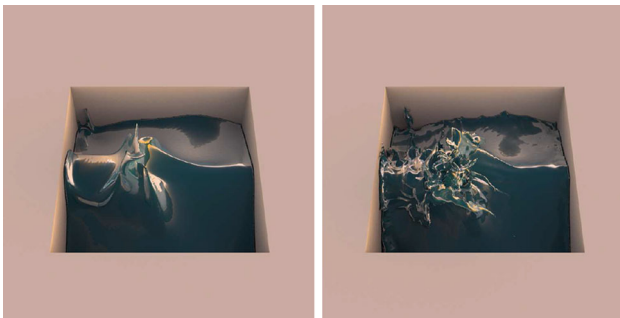


Fig. 8 Comparison of semi-Lagrangian advection and our method. Though the level sets were advected under the same velocity field, our method was able to capture many small details of the water surface

more efficient using multi-thread techniques. We performed our DBCM method for all simulations with multi-thread (8 threads) and it is quite efficient.

We summarize some detailed timing of examples presented above in Table 1 (N_b refer to the resolution of base simulation). With parallel computation, our DBCM method is fast enough for applications in fluid simulation.

The computation time for BFECC advection of χ and RK4 advection of χ^R depends on the resolution of the grids while that of error estimation also depends on the sizing functions in specific applications. As we applied our DBCM method to single-threaded fluid simulators, most of the computation time was spent on the fluid solver itself, whose accelerations are beyond the scope of this paper.

5.4 Timing

We maintain the advection of χ with BFECC method and careful remapping scheme. Indeed, BFECC is slower than semi-Lagrangian method and the error estimation also requires extra expenses.

However, our mapping function can be used for arbitrary set functions. The more set functions we used, the more we can benefit from our method. Besides, we have proposed an importance sampling method for efficient error estimation. Finally, the method is highly parallelizable so we can make it

6 Limitations and future work

Though we have shown the properties of our method for accurate advection, there exist some limitations still. The mapping functions are maintained on regular grids, which occupy quite a lot of memories. This limits the maximum resolution that we can use. Adaptive grids may be adopted to make it more memory efficient. The details captured by our method sometimes appear to be a little noisy near the boundary of the water surface. The current method opens a wealth

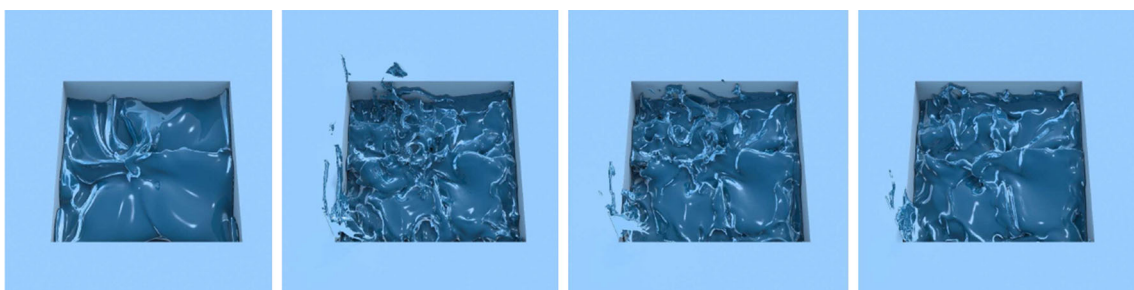


Fig. 9 Dam breaking simulation by the combination of SLC with our DBCM method. Note the thin sheets captured by our method. The resolution of base fluid simulation was 64^3 . N_c was 128 and N_f was 256 for DBCM method

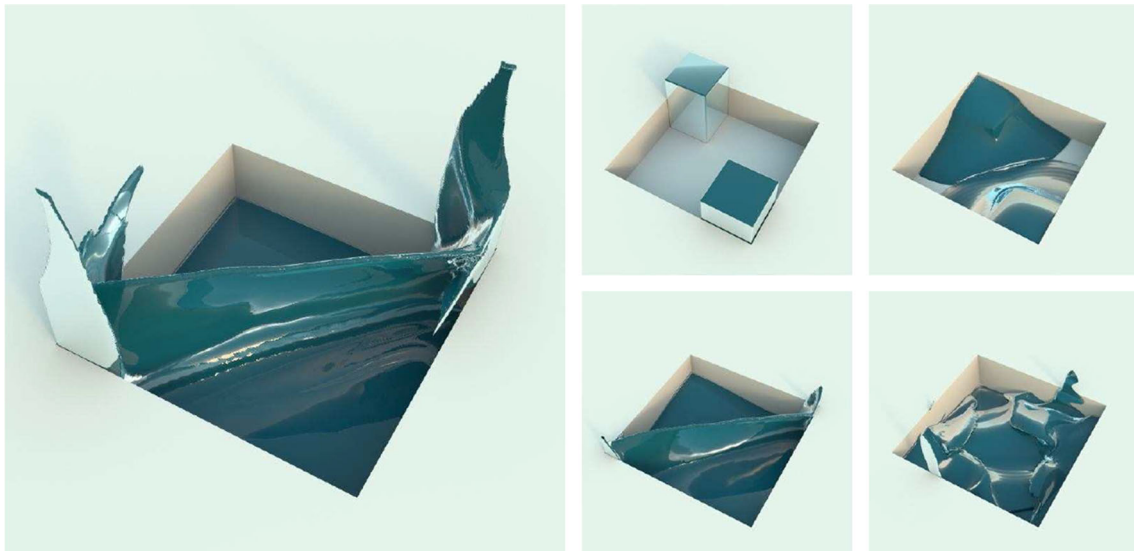


Fig. 10 Double-dam breaking simulation by the combination of SLC with our method. Two columns of water were released to collide with each other to create tall thin sheets and then fell down. Our method

captured such details easily without much effort. The resolution of base fluid simulation was 64^3 . N_c was 128 and N_f was 256.

Table 1 Average computation time (s/step) of our algorithm in selected examples: smoke simulation in Fig. 5; A water bunny released into pool in Fig. 7; Dam breaking in Fig. 9

Example	N_c	N_f	N_b	Dynamic BFECC Characteristic Mapping			Total time
				BFECC advection	RK4 advection	Error estimation	
Smoke	256	256	256	1.88	1.16	1.98	43.59
Water bunny	128	256	128	0.26	1.14	0.78	25.13
Dam breaking	128	256	64	0.25	1.17	0.27	4.72

We used multi-threaded computation for DBCM while the fluid simulation was left to single-threaded. Timing was measured in steps. Timing for each frame may take longer, which depends on the CFL timestep of the simulation

of possibilities of applications. We have only applied it to the advection problem of scalar fields in this paper. We hope to incorporate it into the advection of a vector field, such as velocity directly, which would be an interesting subject.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This research is supported by NSFC Grant (61272326), HK RGC Grant (416212) and the grant of University of Macau.

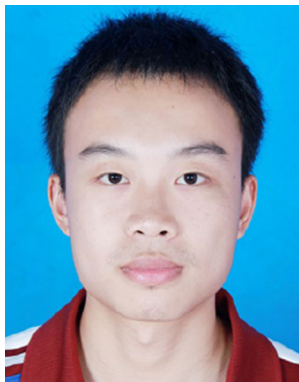
References

- Bargteil, A.W., Goktekin, T.G., O'Brien, J.F., Strain, J.A.: A semi-lagrangian contouring method for fluid simulation. *ACM Trans. Graph.* **25**(1), 19–38 (2006)
- Dupont, T.F., Liu, Y.: Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *J. Comput. Phys.* **190**(1), 311–324 (2003)
- Enright, D., Fedkiw, R., Ferziger, J., Mitchell, I.: A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* **183**(1), 83–116 (2002)
- Enright, D., Marschner, S., Fedkiw, R.: Animation and rendering of complex water surfaces. *ACM Trans. Graph.* **21**(3), 736–744 (2002)
- Fedkiw, R., Stam, J., Jensen, H.W.: Visual simulation of smoke. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '01*, pp. 15–22. ACM, New York (2001)
- Hachisuka, T.: Combined lagrangian–eulerian approach for accurate advection. In: *ACM SIGGRAPH 2005 Posters, SIGGRAPH '05*. ACM, New York (2005)
- Kim, B., Liu, Y., Llamas, I., Rossignac, J.: Flowfixer: Using bfecc for fluid simulation. *Proceedings of the First Eurographics Conference on Natural Phenomena. NPH'05*, pp. 51–56. Eurographics Association, Aire-la-Ville (2005)
- Kim, B., Liu, Y., Llamas, I., Rossignac, J.: Advections with significantly reduced dissipation and diffusion. *IEEE Trans. Vis. Comput. Graph.* **13**(1), 135–144 (2007)
- Kim, D., Song, O.Y., Ko, H.S.: A semi-lagrangian cip fluid solver without dimensional splitting. *Comput. Graph. Forum* **27**(2), 467–475 (2008)
- Mercier, O., Nave, J.C.: The characteristic mapping method for the linear advection of arbitrary sets. [arXiv:1309.2731](https://arxiv.org/abs/1309.2731) (2013)
- Nave, J.C., Rosales, R.R., Seibold, B.: A gradient-augmented level set method with an optimally local, coherent advection scheme. *J. Comput. Phys.* **229**(10), 3802–3827 (2010)

12. Pharr, M., Humphreys, G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (2010)
13. Selle, A., Fedkiw, R., Kim, B., Liu, Y., Rossignac, J.: An unconditionally stable maccormack method. *J. Sci. Comput.* **35**(2–3), 350–371 (2008)
14. Selle, A., Rasmussen, N., Fedkiw, R.: A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.* **24**(3), 910–914 (2005)
15. Stam, J.: Stable fluids. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '99*, pp. 121–128. ACM Press/Addison-Wesley Publishing Co., New York (1999)
16. Tessendorf, J., Pelfrey, B.: The characteristic map for fast and efficient vfx fluid simulations. In: *Computer Graphics International Workshop on VFX, Computer Animation, and Stereo Movies*. Ottawa, Canada (2011)
17. Zalesak, S.T.: Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31**, 335–362 (1979)
18. Zhu, Y., Bridson, R.: Animating sand as a fluid. *ACM Trans. Graph.* **24**(3), 965–972 (2005)



Xiaosheng Li received his B.Sc. degree from Sun Yat-Sen University, Guangzhou, China, in 2010. He is currently a Ph.D. candidate at the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include computer graphics, fluid simulation



Le Liu received his B.Sc. degree from Sun Yat-Sen University, Guangzhou, China, in 2011. He is currently a Ph.D. candidate at the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. He majors in Computer Graphics.



Wen Wu received her Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong. She is currently an assistant professor in Department of Computer and Information Science at University of Macau and also an honorary research associate in the Department of Computer Science and Engineering at The Chinese University of Hong Kong. Her research interests include medical simulation, virtual reality and physically based animation.



Xuehui Liu received her B.Sc. and M.Sc. degrees from Xiang Tan University, Hunan, and received her Ph.D. degree in 1998 from the Institute of Software, Chinese Academy of Sciences. Since then she has been working at the Institute of Software, Chinese Academy of Sciences, and is now an Associate Professor. Her research interests include realistic image synthesis, physically based modeling, and simulation.



Enhua Wu received his B.Sc. degree in 1970 from Tsinghua University, Beijing, and his Ph.D. degree in 1984 from University of Manchester, UK. Since 1985, he has been working at the Institute of Software, Chinese Academy of Sciences, and from 1997 he has been also teaching in University of Macau. He is a member of IEEE & ACM, and Fellow CCF. His research interests include realistic image synthesis, virtual reality, physically based modeling, and scientific visualization.