# Advections with Significantly Reduced Dissipation and Diffusion

ByungMoon Kim, Yingjie Liu, Ignacio Llamas, Jarek Rossignac
Georgia Institute of Technology

***Abstract*— Back and Forth Error Compensation and Correction (BFECC) was recently developed for interface computation using a level set method. We show that BFECC can be applied to reduce dissipation and diffusion encountered in a variety of advection steps, such as velocity, smoke density, and image advections on uniform and adaptive grids and on a triangulated surface. BFECC can be implemented trivially as a small modification of the first-order upwind or semi-Lagrangian integration of advection equations. It provides second-order accuracy in both space and time. When applied to level set evolution, BFECC reduces volume loss significantly. We demonstrate the benefits of this approach on image advection and on the simulation of smoke, bubbles in water, and the highly dynamic interaction between water, a solid, and air. We also apply BFECC to dye advection to visualize vector fields.**

***Index Terms*— Advection, Diffusion, Dissipation, Fluid, Smoke**

## I. Introduction

In computer graphics applications, such as fluid simulation and vector field visualization, various properties, including velocity vector components, smoke density, level set values, and texture or dye colors, must often be transported along some vector field. Those transportations, referred to as *advection*, can be performed on various grids such as uniform and adaptive grids or triangulated surfaces. Here are five common uses of advection in computer graphics:

- *Velocity advection* transports the velocity field along the velocity itself. This step is required in all non-steady flow simulation based on the Navier-Stokes equation.
- *Smoke density advection* transports smoke along the velocity field.
- Sometimes, we may want to advect a colored image, which may be thought of as texture or colored smoke. We call this process *image advection*.
- When one uses a level set method [1] to simulate a free surface or a two-phase flow such as a water surface simulation, the level set values must be transported as well. We refer to this process as *level set advection*.
- One may want to visualize a vector field by advecting dye on the vector field. We call this process *dye advection*.

Advection steps can be implemented by an upwind [2], [3] or a semi-Lagrangian [4] method. Due to its stability for large time steps, the latter is often preferred. These two methods can be implemented with various order of accuracies, for example, first, second, and higher-order accuracies. Despite their low accuracy, first-order methods are popular in computer graphics because of their simplicity. However, lack of accuracy in first-order methods results in a significant amount of numerical diffusion and dissipation. For example, in velocity advection, fluid motion is dampened significantly, which may remove small scale and even large scale motions. In smoke density advection, a premature dilution of smoke occurs, preventing simulation of weakly dissipative and diffusive smoke. In level set advection, significant volume loss takes place. In dye advection, diffusion causes blur and the dissipation produces dark patterns or even an early termination of the trajectory.

Researchers have proposed solution to each of these problems. For dye and texture advection, combining first-order advection and particles increased the accuracy [5]. For level set advection, the particle level set method [6] produces little volume loss. For smoke advection, cubic interpolation reduces diffusion and dissipation [7]. For velocity advection, small scale motions can be maintained by adding vorticity [7], [8], by using particles [8], [9]. All of these solutions can be considered as problem-specific enhancements of the first-order advection. We notice that the FLIP method, introduced recently in [10], advects properties with particle, and therefore, may be applied to any advection achieving zero dissipation. However, in advecting dye or smoke from a source, new particles may have to be created as smoke or dye volume grows. In contrast, a purely Eulerian-based high-order advection method can reduce dissipation and diffusion significantly while taking advantage of the simplicity of the Eulerian grid.

There are many such high order methods for improving the accuracy in the advection steps, such as the WENO scheme [11], [12] and the CIP method [13], [14]. Generally speaking, higher order methods are more difficult to implement, in particular for non-uniform and adaptive meshes. Also because the solutions may contain singularities, special treatments are usually necessary. In [6], Enright et. al. made the particle level set method [15] more efficient and easier to implement by using a first order semi-Lagrangian method to compute the level set equation while propagating the particles with higher order methods, which produces high resolution near interface corners. Local mesh refinement near non-smooth regions of the solutions is an effective technique for improving the accuracy. However, it increases the implementation complexity, particularly when a sophisticated underlying numerical scheme is used. There is also a trade-off between the levels of adaptive mesh refinement and the formal order of accuracy of the underlying scheme. We propose to improve on all the issues addressed above and demonstrate the accuracy of our methods for a number of problems. The underlying scheme we use for computing the advections is the “back and forth error compensation and correction” algorithm (BFECC) [16] and

[17]. When applied to the first order semi-Lagrangian Courant-Isaacson-Rees (CIR) scheme, this BFECC method has second order formal accuracy in both time and space. It essentially calls the CIR scheme 3 times during each time step, and thus maintains the benefits of the CIR scheme such as the stability with large time steps, convenience for use in non-uniform meshes, and low cost in computation.

In [16], [17], the authors proposed BFECC as an alternative scheme for interface computations using the level set method and have tested it on the Zalesak's disk problem and simple interface movements with static or constant normal velocity fields on uniform meshes. We have found it beneficial to adapt the method to level set advection in fluid simulations that contain complicated dynamically varying velocity fields. It is also useful to further apply the method to other types of advections.

We apply BFECC to the various advection problems mentioned earlier and show that BFECC provides significant reduction of dissipation and diffusion for all these applications. The ideas presented here were introduced in a workshop paper [18]. Here, we provide a more detailed treatment of the proposed solution, and new examples of applicatoins to dye advection, advections on a triangle mesh, and on an adaptive quad-tree mesh.

## II. Previous Work

In fluid simulation, stability problems in earlier work [19] were successfully remedied [4] by introducing semi-Lagrangian advection and implicit solve for the viscosity term. The pressure projection is also introduced to graphics community to enforce incompressibility of the fluid. This solution is popular for the simulation of incompressible fluids such as smoke [7] and for the more challenging free surface flows [15], [20].

Semi-Lagrangian velocity advection [4] comes with built-in dissipation, *i.e.,* the velocity dissipates quickly since the linear interpolation in the first-order semi-Lagrangian produces large error. In [7], vorticity is added to generate a small scale fluid rolling motion. Recently in [8] and [9], vortex particles are used to transport vortices without loss. In [14], the authors addressed this built-in dissipation problem by increasing advection accuracy. They adopted the constrained interpolation profile(CIP) [13] method, which increases the order of accuracy in space by introducing the derivatives of velocity to build a sub-cell velocity profile. A nice feature of this CIP method is that it is *local* in the sense that only the grid point values of one cell are used in order to update a point value. However, in this CIP method, all components of velocity and their partial derivatives should be advected, increasing the implementation complexity and computation time, especially in 3D. In addition, it is also worth noting that CIP has higher order accuracy in space only. Therefore, high order integration of characteristics is also necessary. In contrast, the BFECC method used here can be implemented more easily and exhibits second-order accuracy in both space and time and is *local* during each of its operational steps.

Song et al. [14] focused on applying CIP to generate more dynamic water surface behavior. We demonstrate that having less dissipative and diffusive advection provides significant benefits in smoke simulations. This is illustrated in the middle five images of Fig. 6, where a large amount of dissipation makes the smoke look dark. In contrast, when BFECC is used, the smoke keeps its full brightness throughout the simulation, shown in the last five images.

The introduction of the level set method to fluid simulation in [20] allows the realistic simulation of fluids with complex free surfaces. The remaining problem was volume loss in the level set method. The solution, known as the particle level set method, proposed subsequently in [15], turned out to be very successful for volume preservation. This method has been broadly used in recent fluid studies, including [6], [8], [21]–[23].

Fluid simulation on a curved surface domain has been the subject of several studies. Recently, [24] introduced unstructured lattice Boltzmann model for fluid simulation on triangulated surfaces. For the advection of smoke, the first-order semi-Lagrangian advection is used on a flattened neighborhood. The author of [25] mapped the surface on a flat domain and then solved the Navier-Stokes equation. The advections still remain first-order. The authors of [26] proposed to do a semi-Lagrangian advection directly on triangular mesh without mapping it onto a flat domain. The accuracy is still first order and dissipation and diffusion cannot be smaller than the amount that already exists in the advection step. We show that BFECC can be easily applied to the advections on a triangulated domain.

For all simulations of free surfaces in this paper, we use the two-phase fluid model and variable density projection, both of which have been broadly studied in mathematics and fluid mechanics [27]–[29], and have been used in graphics applications in [23], [30], where the authors simulated air bubbles rising and merging and in [13], [14], where splash style interactions between water surface and air are studied.

## III. The BFECC Method

In this section, we review the BFECC method introduced in [16]. Since we want to apply it to various advections, we use $\varphi$ to denote an advected quantity and reserve the symbol $\phi$ for the level set function through the presentation of this paper. This $\varphi$ can be the velocity components $u, v, w$, smoke density, the RGB color of an image or a dye, or level set function $\phi$. For a given velocity field $\mathbf{u}$, $\varphi$ satisfies the advection equation

$$\varphi_t + \mathbf{u} \cdot \nabla \varphi = 0. \tag{1}$$

We briefly describe the BFECC method here. Let $L$ be the first-order upwinding or semi-Lagrangian integration steps to integrate (1), *i.e.,*

$$\varphi^{n+1} = L(\mathbf{u}, \varphi^n). \tag{2}$$

The implementation of $L(\cdot,\cdot)$ can be found in [2], [3]. With this notation, the BFECC can be written as follows:

$$\begin{gathered}\varphi^{n+1} = L\left(\mathbf{u}, \quad \varphi^n + \frac{1}{2}\left(\varphi^n - \bar{\varphi}\right)\right) \\ \text{where } \bar{\varphi} = L\left(-\mathbf{u}, \ L\left(\mathbf{u}, \varphi^n\right)\right).\end{gathered} \tag{3}$$

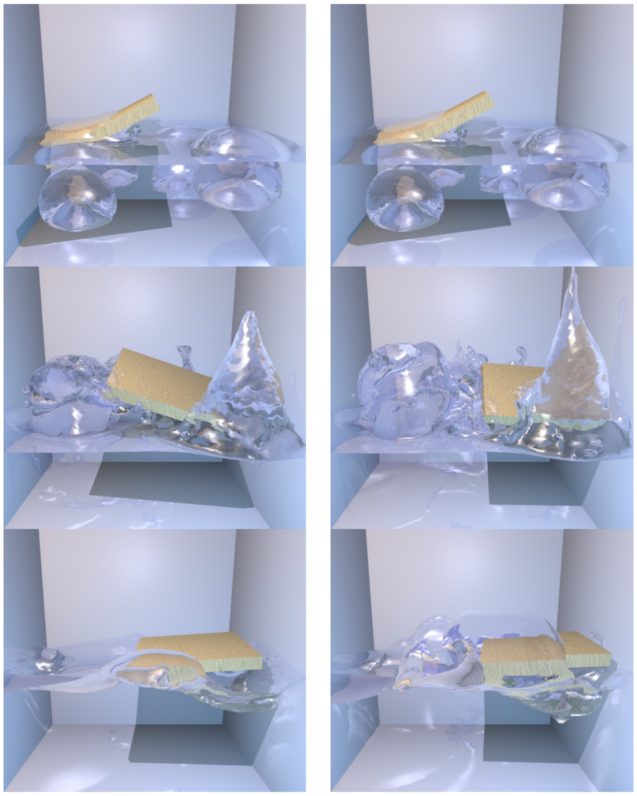

Fig. 1. In the right column, a highly dynamic behavior of water interaction with air, air bubbles, and a solid is made possible by the two-phase formulation and the BFECC-based reduction of the dissipation in the velocity advection step. In the left column, the BFECC is turned off and the splash is reduced. The grid resolution is $81^3$ with $\Delta x = 0.0125m$, and the time step $\Delta t = 0.002sec$.

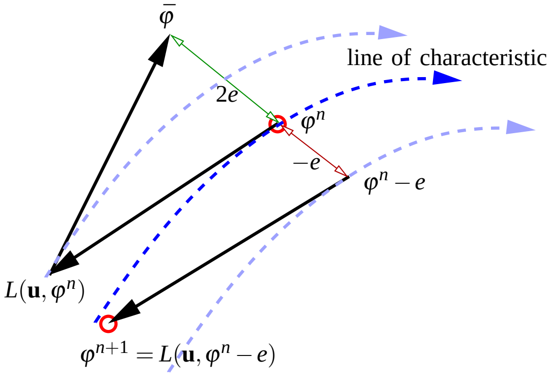
$\bar{\varphi}$
line of characteristic
$2e$
$\varphi^n$
$-e$
$\varphi^n - e$
$L(\mathbf{u}, \varphi^n)$
$\varphi^{n+1} = L(\mathbf{u}, \varphi^n - e)$

Fig. 2. Sketch of computing error by comparing with the forward/backward advected value and then compensating error before the final advection. Notice that this is only a sketch and should not be interpreted geometrically. For example, $e$ is not a vector.

As illustrated in Fig. 2, one may understand this method intuitively as follows. If the advection step $L(\cdot,\cdot)$ is exact, the first two forward and backward steps return the original value, *i.e.*, $\varphi^n = \bar{\varphi}$. However, this is not the case due to the error in advection operation $L$. Suppose $L$ contains error $e$. Then the first two forward and backward steps will produce error $2e$, *i.e.*, $\bar{\varphi} = \phi^n + 2e$. Therefore, the error can be computed as $e = -\frac{1}{2}(\varphi^n - \bar{\varphi})$. We subtract this error $e$ before the final forward advection step. Then the equation (3) becomes $\varphi^{n+1} = L(\mathbf{u}, \varphi^n - e)$. This step will add an additional $e$, which will be cancelled by the subtracted amount $-e$. This method is proven to be second-order accurate in both space and time [16], [17].

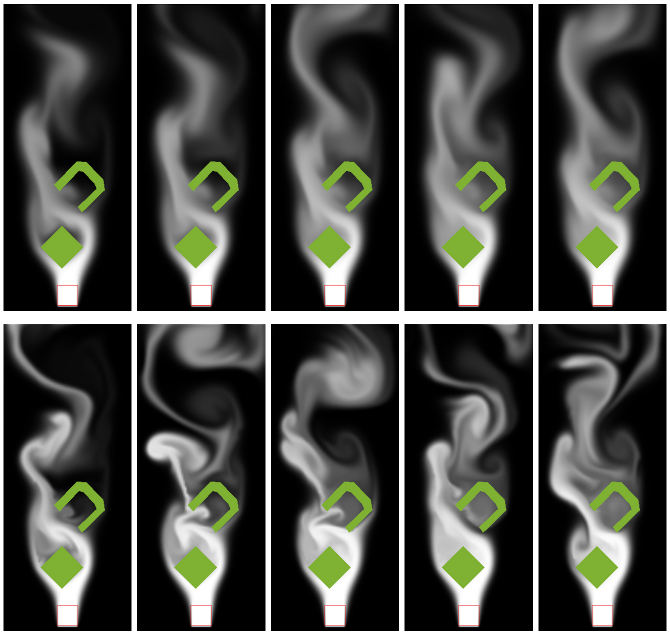

Fig. 3. On the top, we used first-order velocty advection that shows damped fluid motion. On the bottom, we have added the simple BFECC method. Notice the small scale details as well as large scale fluctuations. The grid size is 81×141 with $\Delta x = 0.001m$, and the time step $\Delta t = 0.005sec$.

### A. Implementation of BFECC

In this section, we provide a pseudo-code to demonstrate the simplicity of the BFECC implementation. The function First-Order-Step($u, v, \varphi^n, \varphi^{n+1}$) implements $L(\cdot,\cdot)$, *i.e.*, upwind or semi-Lagrangian integration of the scalar field $\varphi$. Then BFECC is implemented as follows:

```
First-Order-Step($u, v, \varphi^n, \varphi^*$)
First-Order-Step($-u, -v, \varphi^*, \bar{\varphi}$)
$\varphi^* := \varphi^n + (\varphi^n - \bar{\varphi})/2$
First-Order-Step($u, v, \varphi^*, \varphi^{n+1}$)
```

## IV. Applications of BFECC for Various Advections

### A. BFECC for Velocity Advection

We can use (3) to implement the velocity advection step in solving the Navier-Stokes Equation. In this case, $\varphi$ becomes $u, v$, and $w$. We show that BFECC can reduce the damping in the first-order semi-Lagrangian implementation of velocity advection, which is a well-known drawback of semi-Lagrangian advection [4].

For a multiphase flow, this BFECC needs to be turned off near the interface to prevent velocities of different fluids with different densities from mixing, which creates momentum changes. The BFECC may be turned off whenever velocities of different fluids are mixed in backward and forward steps. However, in practice, semi-Lagrangian advections, including

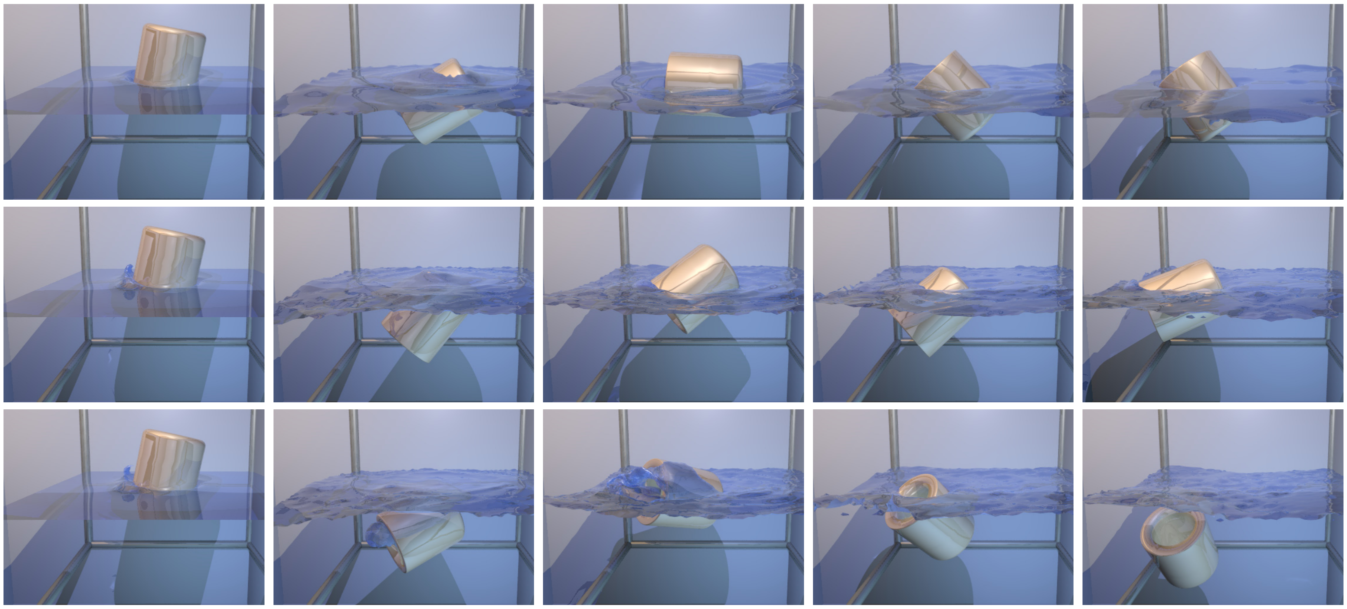

Fig. 4. Simulation of a sinking cup on a $71^3$ grid ($\Delta x = 0.01428m$, $\Delta t = 0.00333sec$). The top row is simulated without the BFECC in both level set and velocity advection steps, where the motion of the cup is damped (the cup does not dive deep into the water) and the detail of the surface is poor. In the center row, this poor surface detail is enriched by turning BFECC on for the level set step, but the cup motion is still damped (the cup goes deeper but it does not tumble). Finally, in the bottom row, the dampening in motion of the cup is remedied by using BFECC for the velocity advection step as well, making the cup sink deeper and tumble as well.

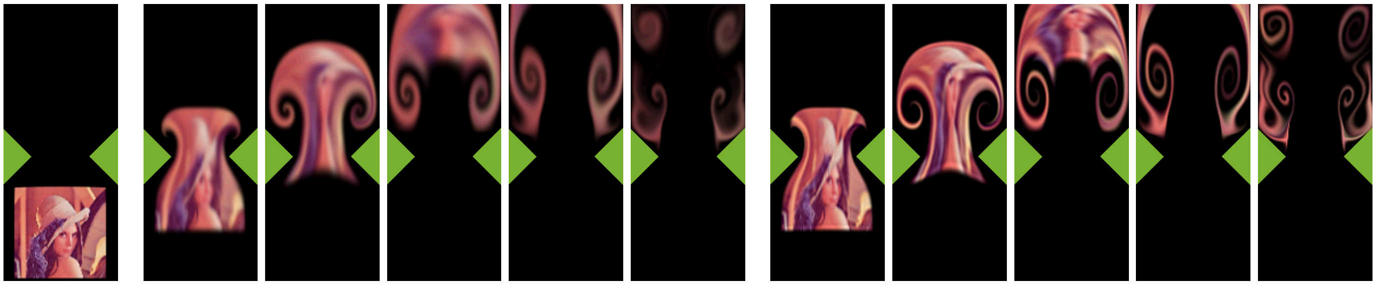

Fig. 5. Advection of an image along with the up-going flow field on 101×251 grid ($\Delta x = 0.0025m, \Delta t = 0.01sec$). The first image shows the initial location of the image. The next six images are computed without the BFECC, where the dissipation/diffusion are significant. The last six images are computed with the BFECC, where the dissipation is greatly reduced and the features of the image can be identified.

BFECC, produces artifacts when the CFL number is greater than $4 \sim 5$. Therefore, the user may need to adjust the time step to meet this requirement. Assuming that the CFL number is less than 5, we simply turn BFECC off, *i.e.*, use the first-order semi-Lagrangian, for the grid points where $|\phi| < 5\Delta x$. For a similar reason, we also turn BFECC off near the boundary of the computational domain. Also notice that this turning-off strategy is only applied for the velocity advection. In other advection applications, we apply BFECC everywhere.

As shown in Fig. 3, applying BFECC adds details as well as large scale fluctuations in smoke motion. Notice that these details and large-scale fluctuations cannot be obtained from the vorticity confinement and vortex particle methods [7], [8], which add only small scale rolling motions. We also performed the same simulation in a coarser grid of 100×40. In this case, the flow did not fluctuate at all around obstacles with first-order semi-Lagrangian advection. However, when BFECC was added, the flow fluctuated as it did in the refined grid. We conclude that BFECC can create physically realistic fluctuations in a coarse grid.

Velocity advection can be important also when rigid bodies are involved. In Fig. 4, the cup does not tumble due to the velocity dissipation in the first-order semi-Lagrangian method, while the cup does tumble when BFECC is applied to the velocity advection step.

### B. BFECC for Smoke Density and Image Advection

We also apply BFECC to the advection of smoke density for smoke simulation. In Figs. 5 and 6, we show that BFECC can reduce dissipation and diffusion significantly. As shown in [16], [17], BFECC is linearly stable in the $l^2$ sense, *i.e.*, $||a||^2_{l^2} = \sum |a_{ij}|^2$ is bounded, when the velocity field is constant, where $a$ is the smoke density. However, density values $a_{ij}$ can become negative or greater than 1.0 for some grid points. In our simulation, the excess amount was negligible, so we simply clamped those values to stay in $[0,1]$.

To measure the diffusion/dissipation amount, we design a test problem similar to the Zalesak's problem. Instead of the notched disk in the Zalesak's problem, we place a color image

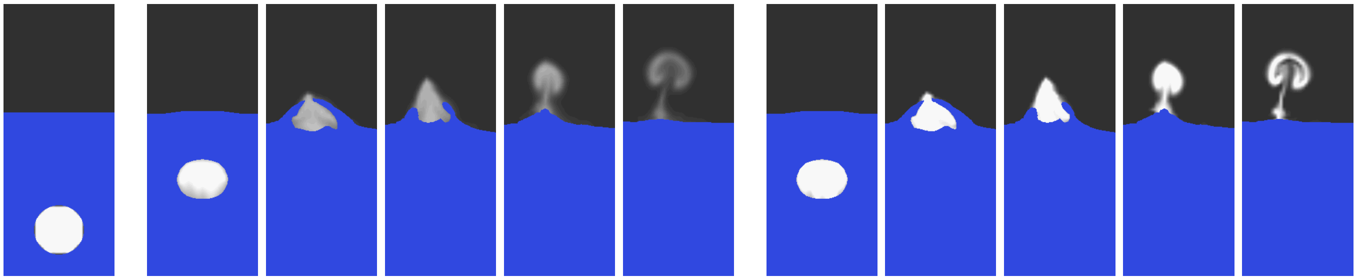

Fig. 6. Simulation of smoke in a bubble rising and bursting on a $41 \times 101$ grid ($\Delta x = 0.0025m, \Delta t = 0.01sec$). The far left image shows the initial bubble. The next five images are without BFECC, where the dissipation/diffusion in the semi-Lagrangian step deteriorate the density of smoke. The last five images simulated with BFECC show significantly reduced dissipation/diffusion, and the smoke is in full density throughout the simulation. All simulation parameters between the two runs are identical, except for the usage of BFECC in smoke advection. Therefore, the only difference is the density of smoke. Also, notice that the simulation time differs by less than 1% since the bulk of the computation time is dominated by the pressure projection step.

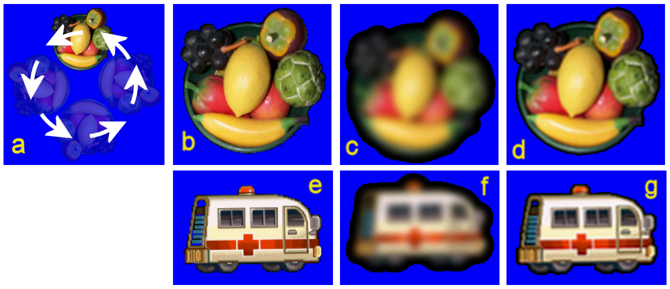

Fig. 7. Test of dissipation and diffusion on an image advection problem along a circular vector field ($801^2$ grid, CFL = 6.29). (b) is the top center portion of the original image (a). (c) is obtained by rotating it 360 degrees using the first-order semi-Lagrangian scheme, where one can see a large amount of dissipation, diffusion, shrinkage of image, and position error. These errors are significantly reduced in (d), where BFECC is used. The blue background region is, in fact, in black, but it is rendered as blue to illustrate the region in which the color is not diffused. (e)-(g) are the same test with another image. Notice the position error in (f) due to the lack of accuracy in time. This is fixed in (g), where BFECC is applied.

and rotate it 360 degrees and then compare it with the original image as shown in Fig. 7. As shown in (d), the dissipation of the color is significantly reduced with BFECC. During advection, the image is also diffused to the neighboring region. To visualize the diffusion amount, we plot background pixels as blue to show the region where the image has been diffused. As shown in (d), the color of the object has almost no diffusion into neighboring region when BFECC is used. Also notice that the position of the image is different from the original location in (f) due to an error in time integration. This is fixed again in (g), where BFECC is used, showing that due to the second-order time accuracy of BFECC, the image follows the vector field more precisely. The computation time was 0.156 seconds (without BFECC) and 0.36 seconds (with BFECC) per frame on a 3GHz Pentuim4.

### C. Dye Advection for Vector Field Visualization

One way to visualize a vector field is to advect a dye on it. A natural approach would be to use the first-order semi-Lagrangian advection method. However, it introduces severe diffusion and dissipation. As shown in the left image of Fig. 8, the result contains a large amount of diffusion and dissipation. This problem has been addressed by several researchers. The authors of [31] applied the level set method to advect the dye without diffusion. Their approach requires the implementation of level set together with a careful treatment to redistancing to prevent volume loss. It also allows only one dye color. In [5], the authors combined semi-Lagrangian and Lagrangian particle advection of dye, which requires an additional non-trivial amount of implementation. In contrast, BFECC requires only a trivial amount of code, and allows a convincing visualization of the vector field, as shown in the right image in Fig. 8.

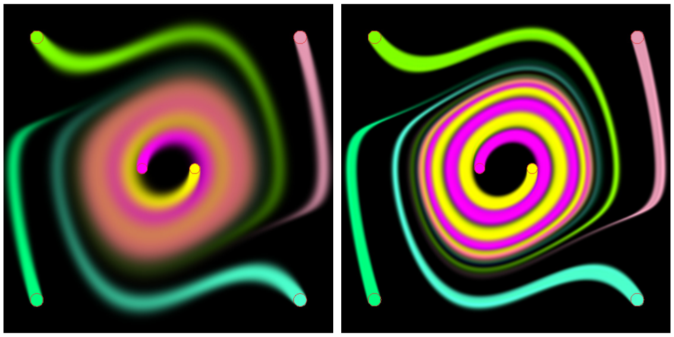

Fig. 8. Visualization of the Van der Pol oscillator on a $401^2$ grid ($\Delta x = 0.000625m, \Delta t = 0.005sec$). The left image shows dye advection with first-order advection. The right image shows dye advection with BFECC, which produces reasonable visualization of the vector field.

### D. BFECC for Level Set Advection

Even though BFECC still has some volume loss in fluid simulation, particularly for small droplets or thin filaments, the performance of BFECC in the fluid simulation is interesting since it is easy to implement and fast. On regular grids, the BFECC approach may not quite adequate for level set advection yet, but its simplicity at least compares well with the particle level set method.

When we use the BFECC for level set advection, *i.e.*, $\varphi = \phi$, redistancing is needed to keep the level set function close to a signed distance function. We use the following redistancing equation [27]

$$\phi_\tau + \mathbf{w} \cdot \nabla\phi = \mathrm{sgn}(\phi) \quad \text{where } \mathbf{w} = \mathrm{sgn}(\phi)\frac{\nabla\phi}{|\nabla\phi|}, \qquad (4)$$

where $\mathbf{w}$ is the velocity vector for redistancing. This equation can be solved by applying the first-order upwinding

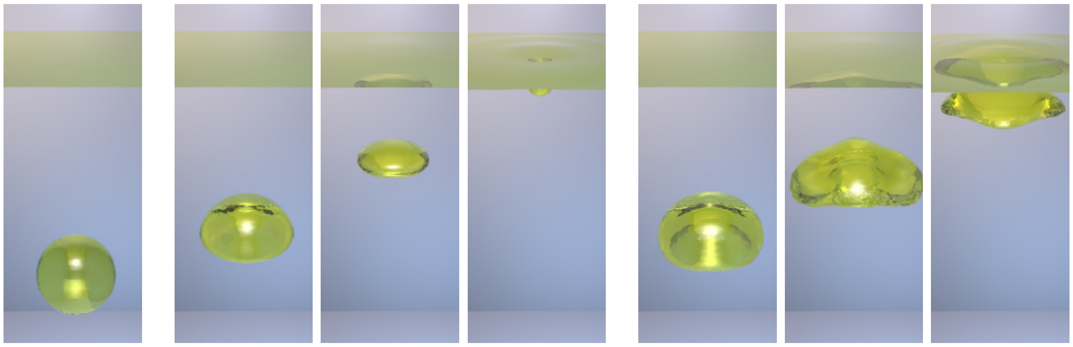

Fig. 9. The far left image shows an air bubble placed in olive oil at time zero. The next three images are first-order semi-Lagrangian implementation of level set advection. The last three images are produced using BFECC and simple redistancing and show significantly reduced volume loss. The grid resolution is $60 \times 100 \times 60$ $(\Delta x = 0.0008333m, \Delta t = 0.001sec)$.

in discretizing the term $\mathbf{w} \cdot \nabla\phi$. An alternative is the semi-Lagrangian style integration, *i.e.*, $\phi^{n+1} = \phi^n(\mathbf{x} - \mathbf{w}\Delta\tau) + \mathrm{sgn}(\phi^n)\Delta\tau$, where $\mathbf{x}$ is the location of each grid point. Hence, $\phi^n(\mathbf{x} - \mathbf{w}\Delta\tau)$ is the $\phi$ value of the previous location. Notice that we do not apply BFECC in the redistancing step.

When these integration formulae for (4) are combined with BFECC, redistancing tends to corrupt good $\phi$ values computed from the second-order accurate BFECC. Thus, if redistancing is turned off near the interface, good $\phi$ values are not corrupted. The conditions in which redistancing is turned off are provided in [16], where significant enhancement was shown in the Zalesak's problem. This simple redistancing is crucial for preserving volume [16], but easy to implement since it simply requires redistancing at points where at least one of the following two conditions is met.

- When the grid point is not close to the interface, *i.e.*, when $\phi_{i,j}$ has the same sign as its eight neighbors.
- When the slope is sufficiently high, *i.e.*, when $|\phi_{i,j} - \phi_{i\pm1,j}| \geq 1.1\Delta x$ or $|\phi_{i,j} - \phi_{i,j\pm1}| \geq 1.1\Delta y$. (5)

### *E. Level Set Advection on Triangulated Surfaces*

We have applied the BFECC to the advection of a scalar field on a triangulated surface. On this surface domain, we explore the advections of a level set. For the first-order semi-Lagrangian advection of this level set, one needs to trace along a vector field defined on a curved surface. The tracing on this curved surface is more complicated than that on a planar domain since the velocity vector should be steered to remain tangent to the surface. On a triangulated piecewise flat domain, this steering occurs when the trajectory moves to a different triangle by crossing an edge or a vertex. The solution to this steering problem is similar to [26], but we explain it in the following way. We perform this edge or vertex crossing steps in a refraction-free manner, i.e., we always proceed in a direction in which the angles of the two sides of the resulting trajectory are identical, as shown in Fig. 10. Using this approach, we can follow the velocity field on the surface, and therefore, we can implement the first-order semi-Lagrangian advection step.

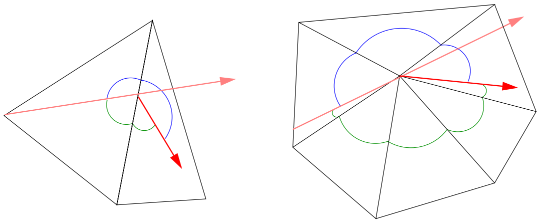

Fig. 10. Steering the directions (from pink to red) so that the angles in the two sides (green and blue) are the same.

Once this first-order advection is implemented, BFECC can be trivially added by calling it three times, as in section III-A. We also implement the simple redistancing strategy similar to (5). Let $\phi_i$ be the level set value at the i th vertex, and $\mathcal{N}_i$ be the set of indices of the vertices neighboring to the i th vertex and let $\mathbf{x}_i$ be the location of i th vertex. The simple redistancing condition on a triangle mesh is

- $\phi_i$ has the same sign as its neighbors.
- $|\phi_i - \phi_j| \geq 1.1||\mathbf{x}_i - \mathbf{x}_j||$, for some $j \in \mathcal{N}_i$.

These two simple schemes reduce smoothing and volume and shape changes of the disk significantly, as shown in Fig. 11.

For fluid simulation on a triangulated surface, one needs to transport a velocity field as well. The velocity vector should always remain on the surface. The authors of [26] also provide a solution to this problem by steering the coordinate frame where the vector is represented. Steering was performed in a way that minimized the twist. Notice that this approach is a discretized version of the parallel transport, which is known as a method for transporting a coordinate frame on manifolds [32]. Using this vector field transport idea, one may implement fluids with free surfaces on a triangulated surface. Since BFECC can significantly improve level set advection as illustrated in Fig. 11, it could be combined with [26] to create a fluid simulator on a triangulated surface.

*Computation of Gradient:* The redistancing wind $\mathbf{w}$ in (4) contains a gradient term $\nabla\phi$. In this section, we show the computation of gradient using a precomputed gradient operator.

Let $\Delta\phi_{ij} = \phi_j - \phi_i$ and let $\Delta x_{ij} = x_j - x_i$, where $x_i$ is the x-coordinate of $\mathbf{x}_i$. Similarly, define $\Delta y_{ij}$ and $\Delta z_{ij}$. Then, we can express the variation of $\phi$ in discrete form.

$$\Delta\phi_{ij} \approx \phi_x \Delta x_{ij} + \phi_y \Delta y_{ij} + \phi_z \Delta z_{ij} \quad , \quad \forall j \in \mathcal{N}_i \qquad (6)$$

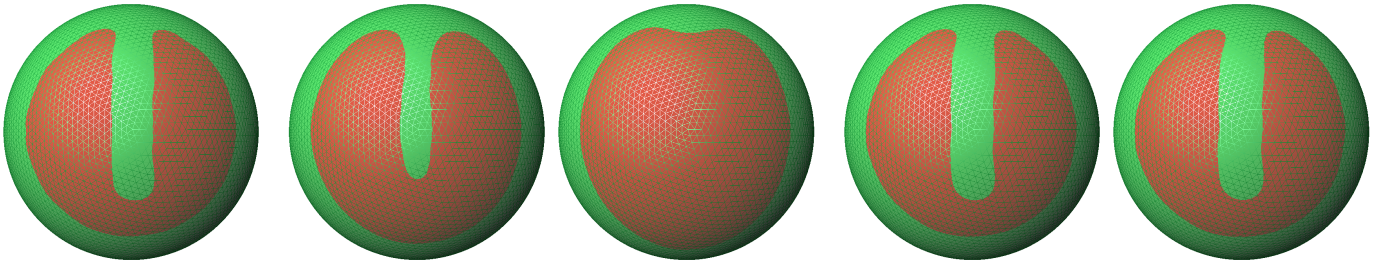

Fig. 11. Advections of the Zalesak's disk on a sphere. The far left image shows an initial disk. The middle two and right two images show the disk after one and two rotations about the vertical axis with first-order advection (middle) and BFECC advection (right). The mesh has 18,000 triangles and 9,002 vertices The time step $\Delta t = 0.01$.

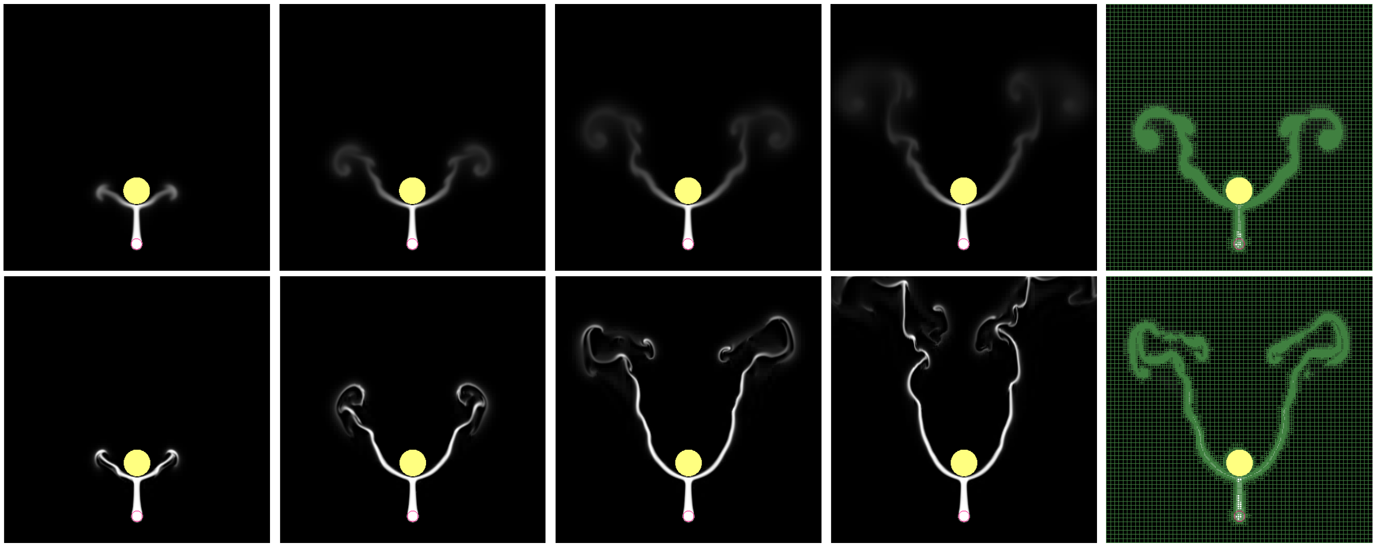

Fig. 12. Simulation of smoke on an adaptive quadtree mesh of maximum resolution $512^2$ ($\Delta t = 0.0001, \Delta x = 0.1m/512$ at maximum resolution). The top row is without BFECC, where the smoke diffusion and dissipation are large. When a lesser diffusive and dissipative smoke is needed, one can trivially implement BFECC and generate a smoke with significantly reduced dissipation and diffusion, as shown in the bottom row, which is simulated with BFECC. The amount of diffusion and dissipation is controlled by adjusting the diffusion and dissipation coefficients in the diffusion and dissipation steps, respectively. Thus, BFECC decouples advection from dissipation and diffusion steps. The last column shows quadtree grids that correspond to the results in the third column.

This system of equation can be written in a martix form.

$$\mathbf{d} = A\nabla\phi \tag{7}$$

$$\text{where} \quad \mathbf{d} = \begin{bmatrix} \Delta\phi_{ij_1} \\ \Delta\phi_{ij_2} \\ \cdots \\ \Delta\phi_{ij_n} \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} \Delta x_{ij_1} & \Delta y_{ij_1} & \Delta z_{ij_1} \\ \Delta x_{ij_2} & \Delta y_{ij_2} & \Delta z_{ij_2} \\ \cdots & \cdots & \cdots \\ \Delta x_{ij_{n_i}} & \Delta y_{ij_{n_i}} & \Delta z_{ij_{n_i}} \end{bmatrix} \tag{8}$$

and $n_i$ is the number of neighbors of i $^{\text{th}}$ vertex. The least square solution of (8) is

$$\nabla\phi = \mathbf{A}^{\dagger}\mathbf{d} \tag{9}$$

where $\mathbf{A}^{\dagger}$ is the Moore-Penrose pseudo inverse of $\mathbf{A}$. Now, suppose we represented $\nabla\phi$ in local coordinate of any two orthonormal tangent vectors $\mathbf{s}_1, \mathbf{s}_2$ and the normal vector $\mathbf{n}$. Let $\mathbf{S} = [\mathbf{s}_1\ \mathbf{s}_2] \in \mathbb{R}^{3\times 2}$. Then, each row-vector of $\mathbf{A}$ can be represented by the $\mathbf{s}_1, \mathbf{s}_2$ and $\mathbf{n}$, *i.e.*,

$$\mathbf{A} = \begin{bmatrix} \mathbf{AS} & \mathbf{An} \end{bmatrix} \begin{bmatrix} \mathbf{S}^T \\ \mathbf{n}^T \end{bmatrix} \quad , \quad \mathbf{AS} \in \mathbb{R}^{n_i \times 2}, \mathbf{An} \in \mathbb{R}^{n_i \times 1} \tag{10}$$

Suppose the surface is locally smooth, then $\mathbf{An}$ is small and can be neglected.

$$\mathbf{A}^{\dagger} \approx \left(\mathbf{AS}\,\mathbf{S}^T\right)^{\dagger} = \mathbf{S}\left(\mathbf{AS}\right)^{\dagger} \tag{11}$$

Finally, $\nabla\phi$ is computed as

$$\begin{aligned} \nabla\phi &\approx \mathbf{S}\ (\mathbf{AS})^{\dagger}\,\mathbf{d} \\ &= \mathbf{S}\ \left(\mathbf{A}_S^T\mathbf{A}_S\right)^{-1}\mathbf{A}_S^T\ \mathbf{d} \quad , \quad \mathbf{A}_S = \mathbf{AS} \end{aligned} \tag{12}$$

Notice that when the mesh is not deforming, $\mathbf{S}\ \left(\mathbf{A}_S^T\mathbf{A}_S\right)^{-1}\mathbf{A}_S^T \in \mathbb{R}^{3\times n_i}$ is constants and can be precomputed as a gradient operator.

### *F. BFECC for Adaptive Mesh*

In an adaptive mesh such as an octree [33], the interpolation required for semi-Lagrangian advection is more complicated to implement. This complexity is often already high in the simplest linear interpolation. Therefore, higher order nonlinear interpolation tends to be much more complex. This superb complexity of high-order interpolation can be easily avoided when one uses BFECC. To verify the applicability of BFECC on adaptive mesh, we implement a smoke simulator on a quad tree mesh similar to [33] and show that it can reduce the diffusion of smoke. Due to little diffusion obtained by BFECC, we can simulate a thin filament of smoke. Since the smoke remains in a thin region, the mesh is refined only in the thin region. Notice that when one uses first-order advection, smoke will diffuse into neighborhood quickly, and a mesh needs to be refined in larger region. The benefits of BFECC on this quadtree mesh is illustrated in Fig. 12.

## V. Additional Discussions

### A. Fluid Simulation Overview

Consider the following Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla \cdot (\nabla \mathbf{u}) - \frac{1}{\rho} \nabla P + \mathbf{f}. \tag{13}$$

We follow the operator splitting steps proposed in [4] except for the advection step, in which we use BFECC, and for the projection step, in which we use the variable density pressure projection. We use the standard staggered grid [7]. Suppose all terms in (13) except for $-\frac{1}{\rho}\nabla P$ are treated, and let the velocity obtained so far be $\tilde{\mathbf{u}}$. The final step is applying the variable density pressure projection step to enforce the continuity equation $\nabla \cdot \mathbf{u} = 0$, i.e., solving the equation $\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla P\right) = \nabla \cdot \tilde{\mathbf{u}}$. Its first order discretization is

$$\begin{aligned} \frac{\Delta t}{\Delta x^2} \left( \frac{P_{i-1,j} - P_{i,j}}{\rho_{i-\frac{1}{2},j}} + \frac{P_{i+1,j} - P_{i,j}}{\rho_{i+\frac{1}{2},j}} + \frac{P_{i,j-1} - P_{i,j}}{\rho_{i,j-\frac{1}{2}}} + \frac{P_{i,j+1} - P_{i,j}}{\rho_{i,j+\frac{1}{2}}} \right) \\ = \frac{1}{\Delta x} \left( \tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j} \ + \ \tilde{v}_{i,j+\frac{1}{2}} - \tilde{v}_{i,j-\frac{1}{2}} \right). \end{aligned} \tag{14}$$

We assume $\Delta x = \Delta y$ here and through the rest of the presentation. The extension to 3D is straightforward and hence omitted. This first order approximation is identical to [14]. Higher-order formulations can be found in [34], [35]. If $\rho$ is constant, we have the pressure projection $\frac{\Delta t}{\rho} \nabla^2 P = \nabla \cdot \tilde{\mathbf{u}}$ introduced in [4]. We also include a simple implementation of surface tension similar to [35].

### B. Notes on Boundary Conditions and CFL Number

In simulation of liquid, only ceiling is open and all other boundaries (bottom, left, right, front, and back) are closed so that the water does not flow outside the domain. For smoke simulation, the top and bottom are open to create an upwaed air flow. We can open or close walls by choosing the proper boundary conditions in the pressure projection step. For open walls, we used the Dirichlet condition with an ambient pressure value of zero, and for closed walls, we used the Neumann condition with zero slope (and the same pressure).

For velocity advection, we used Dirichlet condition (zero) for closed wall and Neumann condition (zero slope) for open boundary. For smoke, and image advections, we used zero (Drichlet) smoke density, and image color for all boundaries. For level set advection, we used Neumann condition with zero-slope, i.e., whenever we need a value outside the computational domain, we used the value at the point closest to the domain.

Even though semi-Lagrangian advection is unconditionally stable, BFECC applied to semi-Lagrangian advection produces grid artifacts when CFL number is larger than 4∼5. Therefore, we choose a small enough time step so that the CFL number does not exceed this bound for most of the time. We did not perform adaptive time stepping to strictly ensure CFL bound. Instead, we simply choose small enough time steps, which are provided in figure captions.

### C. Discussions

We test BFECC in different fluid simulations. We simulate air-water and olive oil-air interactions, whose properties are provided in Table I. Water is rendered in a bluish color, and olive oil is rendered in a yellowish color. We use the PovRay (http://povray.org) to render images.

|  | $\rho[kg/m^3]$ | $\nu[m^2/sec]$ | Surf. Tension [N/m] |
|---|---|---|---|
| air | 1.125 | $1.7\times10^{-5}$ |  |
| water | 1000 | $1.0\times10^{-6}$ | 0.07 |
| olive oil | 910 | $9.2\times10^{-5}$ | 0.035 |

TABLE I

Properties of fluids used in simulations

In Fig. 4, we simulated interactions between a cup, air, and water. The cup is released upside down near the water surface. Due to its weight, the cup sinks deep into water, but it soon rises again because of the air in it. However, in the top, where we turned BFECC off for velocity advection, the water is dissipative, preventing the cup from tumbling. In the bottom, we use BFECC for velocity advection where the velocity dissipation is small, and hence, the cup can tumble 180 degrees. This example indicates that reducing velocity dissipation could be important in simulating fluid and rigid body interactions.

We implement the *rigid fluid* method [21] to simulate rigid body and fluid interaction in Figs. 1 and 4, where buoyancy is automatically obtained by applying variable density projection similar to [36]. We use multiple pressure projections to address the seeping problem mentioned in [21]. We found that the angular momentum of the rigid body tends to be reduced per projection. Therefore, we reinforce the angular momentum per each projection step. This multi-projection treatment is slow but easy to implement. The effect of this simple multi-projection approach is illustrated in Fig. 13. Notice that the seeping problem was solved with only requiring 2 projection steps (one to compute the coupling force and one to ensure water does not leak into the solid) in [36].

The computation time varies with the complexity of the fluid motions. In a simple bubble rising situation without a rigid body, it took a few seconds per time step using $50^3$ mesh. The cup example in Fig. 4 has multiple pressure projections, taking about 30 to 130 seconds per time step on a $70^3$ grid.

## VI. Conclusion

We have shown that the BFECC scheme can be used to improve various advection steps. Once the simple first-order upwinding or semi-Lagrangian steps for velocity, smoke density, image, dye, or level set advections are implemented, BFECC can be added with a trivial amount of code. We show that this simple extension yields significantly reduced diffusion and dissipation in the four advection steps of the fluid simulation, including two-phase flows and rigid bodies and in dye advection for vector field visualization. We also show BFECC is valid for various domains such as uniform, quadtree, and triangulated domains. The benefits of the proposed approach are illustrated in the accompanying video.

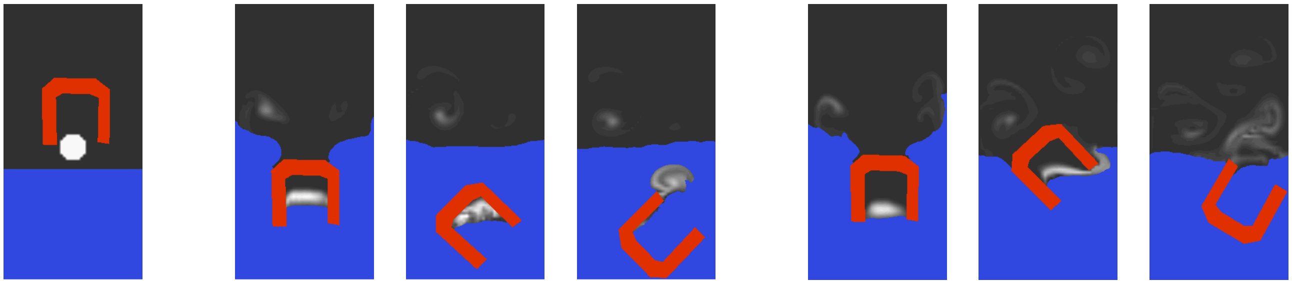

Fig. 13. Dropping a cup into water on a $51\times101$ grid ($\Delta x = 0.01m, \Delta t = 0.005sec$). The far left image shows the initial configuration. The three images in the middle are with two projections and show a significant amount of air lost. The three images on the right are with nine projections. They show that enough air is trapped inside the cup, causing the heavy cup ($\rho = 1300kg/m^3$) to rise again. Also, notice the patterns of the smoke that are diffused or dissipated little, thanks to the smoke advection, using BFECC in both cases.

## VII. Acknowledgement

This work was supported by the NSF under the ITR Digital clay grant 0121663 and the NSF DMS 0511815.

## References

[1] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[2] S. J. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.

[3] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[4] J. Stam, "Stable fluids," in *SIGGRAPH*. ACM, 1999, pp. 121–128.

[5] B. Jobard, G. Erlebacher, and M. Y. Hussaini, "Lagrangian-eulerian advection of noise and dye textures for unsteady flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, 2002.

[6] D. Enright, F. Losasso, and R. Fedkiw, "A fast and accurate semi-lagrangian particle level set method," *Computers and Structures*, vol. 83, pp. 479–490, 2005.

[7] R. Fedkiw, J. Stam, and H. Jensen, "Visual simulation of smoke," in *SIGGRAPH*. ACM, 2001, pp. 23–30.

[8] A. Selle, N. Rasmussen, and R. Fedkiw, "A vortex particle method for smoke, water and explosions," in *SIGGRAPH*. ACM, 2005.

[9] S. I. Park and M. J. Kim, "Vortex fluid for gaseous phenomena," in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005.

[10] Y. Zhu and R. Bridson, "Animating sand as a fluid," in *SIGGRAPH*. ACM, 2005.

[11] X. D. Liu, S. Osher, and T. Chan, "Weighted essentially non-oscillatory schemes," *Journal of Computational Physics*, vol. 115, no. 1, pp. 200–212, 1994.

[12] G. Jiang and C.-W. Shu, "Efficient implementation of weighted ENO schemes," *Journal of Computational Physics*, vol. 126, pp. 202–228, 1996.

[13] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki, "Realistic animation of fluid with splash and foam," in *EUROGRAPHICS*, vol. 22, 2003.

[14] O. Song, H. Shin, and H. Ko, "Stable but nondissipative water," *ACM Transactions on Graphics*, vol. 24, no. 1, pp. 81–97, 2005.

[15] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," in *SIGGRAPH*. ACM, 2002.

[16] T. F. Dupont and Y. Liu, "Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function," *Journal of Computational Physics*, vol. 190, no. 1, pp. 311–324, 2003.

[17] ——, "Back and forth error compensation and correction methods for semi-lagrangian schemes with application to level set interface computations," *Math. Comp., To appear*, 2006.

[18] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, "Flowfixer: Using bfecc for fluid simulation," in *Eurographics Workshop on Natural Phenomena*, 2005.

[19] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graphical Models and Image Processing*, vol. 58, no. 5, pp. 471–483, 1996.

[20] N. Foster and R. Fedkiw, "Practical animation of liquids," in *SIGGRAPH*. ACM, 2001, pp. 15–22.

[21] M. Carlson, P. J. Mucha, and G. Turk, "Rigid fluid: Animating the interplay between rigid bodies and fluid," in *SIGGRAPH*. ACM, 2004.

[22] T. G. Goktekin, A. W. Bargteil, and J. F. O'Brien, "A method for animating viscoelastic fluids," in *SIGGRAPH*. ACM, 2004, pp. 463–468.

[23] J.-M. Hong and C.-H. Kim, "Discontinuous fluids," in *SIGGRAPH*. ACM, 2005.

[24] Z. Fan, Y. Zhao, A. Kaufman, and Y. He, "Adapted unstructured lbm for flow simulation on curved surfaces," in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005.

[25] J. Stam, "Flows on surfaces of arbitrary topology," in *SIGGRAPH*. ACM, 2003, pp. 724–731.

[26] L. Shi and Y. Yu, "Inviscid and incompressible fluid simulation on triangle meshes," *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, pp. 173–181, 2004.

[27] M. Sussman, P. Smereka, and S. Osher, "A levelset approach for computing solutions to incompressible two-phase flow," *Journal of Computational Physics*, vol. 114, no. 1, pp. 146–159, 1994.

[28] M. Oevermann, R. Klein, M. Berger, and J. Goodman, "A projection method for two-phase incompressible flow with surface tension and sharp interface resolution," Konrad-Zuse-Zentrum für Informationstechnik Berlin, Tech. Rep. ZIB-Report 00-17, 2000.

[29] H. Haario, Z. Korotkaya, P. Luukka, and A. Smolianski, "Computational modelling of complex phenomena in bubble dynamics: Vortex shedding and bubble swarms," in *Proceedings of ECCOMAS 2004*, 2004.

[30] J.-M. Hong and C.-H. Kim, "Animation of bubbles in liquid," in *EUROGRAPHICS*, vol. 22, 2003.

[31] D. Weiskopf, "Dye advection without the blur: A level-set approach for texture-based visualization of unsteady flow," in *EUROGRAPHICS*, vol. 23, 2004.

[32] M. P. do Carmo, *Riemannian Geometry*. Birkhauser, 1994.

[33] F. Losasso, F. Gibou, and R. Fedkiw, "Simulating water and smoke with an octree data structure," in *SIGGRAPH*. ACM, 2004, pp. 457–462.

[34] A. S. Almgren, J. B. Bell, and W. G. Szymczak, "A numerical method for the incompressible navier-stokes equations based on an approximate projection," *SIAM Journal of Scientific Computing*, vol. 17, no. 2, March 1996.

[35] M. Sussman, A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome, "An adaptive level set approach for incompressible two-phase flow," *Journal of Computational Physics*, vol. 148, pp. 81–124, 1999.

[36] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw, "Coupling water and smoke to thin deformable and rigid shells," in *SIGGRAPH*. ACM, 2005.