Fast Animating of Fluid Simulation

Jiaotuan Wang, Xubo Yang and Wenlong Lu
Shanghai Jiao Tong University, Shanghai
E-mail: {wangjiaotuan, xubo.yang, wenwestern}@gmail.com Tel: +86-15216712383

***Abstract*—Existing grid-based techniques cannot suit for high-resolution fluid simulation due to the high computational cost. This paper presents a novel modified linear interpolation(MLI) upsampling method for animating details-abundant fluids quickly. Our method runs on a high-resolution grid while creating a divergence-free velocity field on a coarse grid to speed up the calculation. Different from previous work, we use a novel MLI-upsampling operator to reconstruct a fine velocity field from the coarse divergence-free velocity field. The upsampled velocity field is kept divergence-free without solving Poisson equation on the fine grid. We provide several examples to demonstrate that our method is efficient and effective, and it is ideal for CG application where visual detail and speed are essential.**

I. Introduction

The grid-based technique was introduced to Computer Graphics by Foster and Metaxas [1] in 1997. Next, Stam [2] successfully remedied it by introducing the pressure projection scheme to enforce incompressibility and the semi-Lagrangian algorithm of the advection to ensure the stability of the fluid. The integrated solution is prevalent for the simulation of incompressible fluids, such as smoke [3], water [4] and sand [5], etc. Although many authors such as Fedkiw [3, 4], Bridson [5] have used grid-based techniques to produce visually compelling results, the quality of simulation is still tightly related to the size of the grid, which was limited by the amount of computational resource available.

In this paper, we propose a fast method for simulating fluid with small scale details by a combined fine-coarse grid, using a novel modified linear interpolation(MLI) upsampling as our core tool. We advect the fluid on the fine grid, followed by mapping, coarsen projection step, plus a MLI-upsampling process to reconstruct the fine fluid field. Our upsampling operator maps velocity field from the coarse grid to the fine grid and keeps the divergence-free property naturally for incompressible fluids, which are important phenomena for fluid animation.

Our method can be regarded as a fast simulation of the traditional grid-based approach. As we know, there are three steps to solve Navier-Stokes equations: advection, body force and pressure projection. Among of these steps, solving pressure equation (or we called projection) is the most important but so expensive that the computational cost occupies more than 90% of the traditional fluid solver. Our method can greatly reduce the projection cost by using a novel projection algorithm, and it also produces the same level of details as the traditional high-resolution solver does.

II. Related Work

There are many efforts to enhance the grid-based fluid solver, among which some work tried to use higher order interpolation method in the solver, such as using cubic interpolation instead of linear interpolation, tracing back curved path rather than straight line path [3] in the semi-Lagrangian step, using higher order methods in space, such as BFECC [6], QUICK [7], MacCormack [8], or in time, such as Runge Kutta [9]. Although these methods can decrease the numerical dissipation in traditionally stable fluid solver and enhance details, it involves more computational cost and more complicated implementation.

Some previous work synthesized details by introducing with noise. For example, Kolmolgorov noise [10, 11] and curl noise [12]. Kim [13] proposed a wavelet turbulence model to synthesize different levels of details onto the result of a coarse grid simulator. Energy transport model, introduced by Narain [14], was proposed to track the dynamics of turbulent energy over time and combine it with procedural synthesis process on a base simulator. All of these techniques are successful at adding details but nonphysical, and can produce significantly less realistic results than simply simulating with a higher resolution grid.

Lentine et al. [15] proposed a method to accelerate projection step on fine-grid simulation for incompressible flow. Instead of solving an expensive large Poisson equation on the fine grid, they accelerate simulation by solving one coarse grid Poisson equation and many smaller Poisson equations for the refined cells. Our work is inspired by their research and we adopt a similar framework combining fine and coarse grids to accelerate projection. The main difference is that, by taking advantage of the mass-conserving property of our novel MLI-upsampling operator, we do not solve Poisson equation on the fine grid, and only solve one Poisson equation on the coarse grid in each time step.

III. Method Overview

The motion of fluid is calculated by the numerical analysis of the following incompressible Navier-Stokes equations.

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{f} \tag{1}$$

$$\nabla \cdot \vec{u} = 0 \tag{2}$$

where $\vec{u}$ is velocity, t is time, ρ is density, p is pressure, and $\vec{f}$ is external force. ∇ is vector gradient operator and $\nabla\cdot$ is vector divergence operator.

The classical grid-based method solves these equations in three steps, including advection, body force and projection. More details are given in [16]. The part we are interested in is that, the projection step is most computationally costly while solved by traditional method, which occupies more than 90%.

Therefore, we present a novel projection to replace the traditional projection. As illustrated in Fig. 1, there are three key components in our method: map the velocity field to coarse uniform grid; make the coarse grid velocity field divergence free; and map these velocities back to our fine simulation grid which still keeps the fine velocity field divergence free. Each of them will be discussed in detail as following.

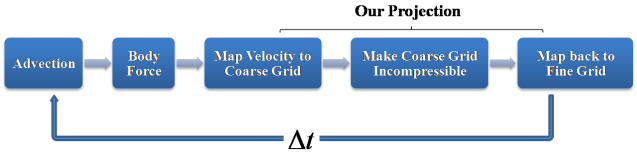

Fig. 1. The entire fluid simulation pipeline.

IV. Our Projection

After steps of advection and body force, we get a fine grid velocity field $\vec{u}^f$. With standard grid-based method, the following projection step will make $\vec{u}^f$ divergence free by solving a costly Poisson equation and updating to an incompressible field $\vec{u}^{n+1}$. However, in this section, we significant reduce the cost of projection step using three steps: map $\vec{u}^f$ to coarse grid to get the coarse velocity $\vec{u}^c$; make $\vec{u}^c$ divergence free via solving a coarse Poisson equation to get a coarse incompressible velocity $\vec{u}^*$; and map $\vec{u}^*$ back to fine grid using MLI-upsamling technique that maintains the incompressibility from coarse to fine.

A. Mapping Velocity to Coarse Grid

In this section, we present a normalized spline kernel method that can map the intermediate fine grid velocity $\vec{u}^f$ to coarse grid and then get coarse velocity $\vec{u}^c$. The normalized spline kernel [17] is popular in particle-based method because this kernel has second order interpolation errors. In this paper, we extend this kernel to grid-based fluid and apply it as our mapping method.

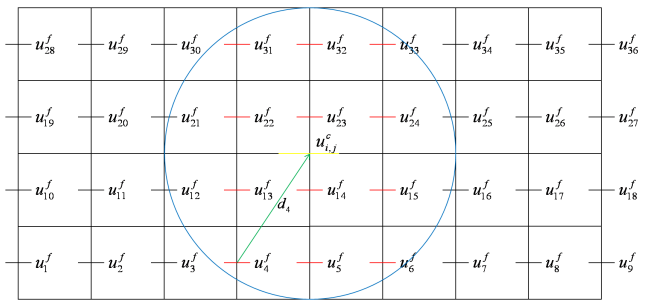

Fig. 2. Mapping the fine velocity to coarse grid.

To map the velocity field $\vec{u}^f$ stored on the fine MAC grid to coarse MAC grid field $\vec{u}^c$, as illustrated in Fig. 2, the result of x component velocity $u^c_{i,j}$ is mapped from fine to coarse by a distance-weighted average function given by:

$$u^c_{i,j} = \frac{\sum_m u^f W(d_m, h)}{\sum_m W(d_m, h)} \tag{3}$$

where $d_m = \|x^f_m - x^c_{i,j}\|$ is the distance between the position of the m^{th} fine grid velocity u^f_m and the position of coarse grid velocity $u^c_{i,j}$, and we choose the normalized spline kernel W as:

$$W(d,h) = \begin{cases} \frac{315}{64\pi h^9}(h^2 - d^2)^3 & , d < \text{h} \\ 0 & , d \geq \text{h} \end{cases} \tag{4}$$

where the influence radius h we adopted is a constant value as 0.5 times the size of a coarse grid.

B. Making Coarse Grid Incompressible

Once we have the downsampled velocity field $\vec{u}^c$ on the coarse grid, we can make this field divergence-free by using a coarse projection operation. Since we define our solid objects on fine grid, standard coarse projection was not exactly accurate, especially around objects. Therefore, we must modify the coarse projection step when handling object boundaries.

We use the variational framework [9] to handle the boundary mismatch between the coarse grid and the fine grid. Instead of using the standard discretization of the laplacian operator, we use the mass-weighted 7-point laplacian stencil to discretize the Poisson equation. We use Preconditioned Conjugate Gradient method to solve the resultant linear system and get pressure result.

Then, we subtract off the pressure gradient from $\vec{u}^c$:

$$\vec{u}^* = \vec{u}^c - \Delta t \frac{1}{\rho} \nabla p \tag{5}$$

where the result $\vec{u}^*$ satisfies the incompressibility of fluid on the coarse grid.

C. Mapping back to Fine Grid

After we get the divergence free velocity field $\vec{u}^*$ on the coarse grid, we need to map this field back to the fine grid. In this section, we present a novel MLI-upsampling method to reconstruct a fine divergence-free field from a coarse divergence-free field.

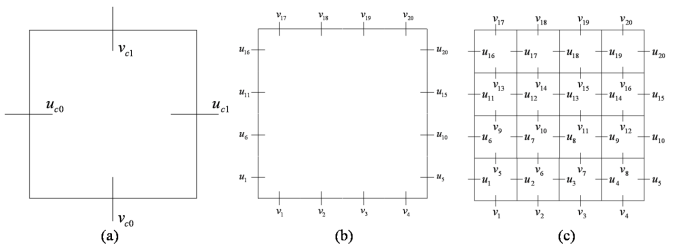

Fig. 3. Reconstructing of the divergence-free field using MLI-upsampling.

Subplot (a), (b) and (c) in Fig.3 show the steps of our MLI-upsampling method. Starting with a coarse grid shown in (a), we need to refine the grid by a factor of 4, as shown in subplot (c). First, from the known values u_{c0} , u_{c1} , v_{c0} , v_{c1} of the coarse faces, we set the boundary face of the fine grid that overlapping the coarse grid equal to the corresponding value of coarse grid, as shown in subplot (b). Then, other values on fine grid are calculated by using a simple linear interpolation of the known values that computed in step (b), as in subplot (c). The MLI-upsampling method also works with other factor m (=2, 3, 4).

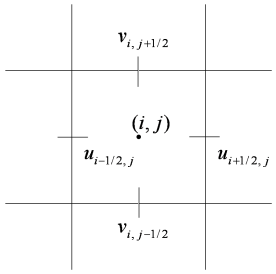

Fig. 4. The reconstruction field refined by a factor of m.

Next, we simply demonstrate that MLI-upsampling method is perfectly satisfying the incompressibility of fluid. Suppose we refine the cell by a factor of m , without loss of generality, we focus on the grid cell (i, j) of the refined cell. As illustrated in Fig. 4, from the known values u_{c0} , u_{c1} , v_{c0} , v_{c1} of the coarse grid, we can calculate the values on the grid cell (i, j) on the fine cell by using MLI-upsampling method, and these values are:

$$u_{i-1/2,j} = \frac{(m-i)u_{c0} + iu_{c1}}{m} \tag{6}$$

$$u_{i+1/2,j} = \frac{(m-i-1)u_{c0} + (i+1)u_{c1}}{m} \tag{7}$$

$$v_{i,j-1/2} = \frac{(m-j)v_{c0} + jv_{c1}}{m} \tag{8}$$

$$u_{i,j+1/2} = \frac{(m-j-1)v_{c0} + (j+1)v_{c1}}{m} \tag{9}$$

Remember the discrete divergence in two dimensions is:

$$(\nabla \cdot \vec{u})_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta x} \tag{10}$$

We simply substitute the values on equations (6-9) into the divergence formula, equation (10), and get the following equation:

$$(\nabla \cdot \vec{u})_{i,j} = \frac{1}{\Delta x}(\frac{u_{c1} - u_{c0}}{m} + \frac{v_{c1} - v_{c0}}{m}) \tag{11}$$

Notice that the coarse grid velocity field is divergence free, so the divergence on coarse grid is:

$$\frac{u_{c1} - u_{c0}}{m\Delta x} + \frac{v_{c1} - v_{c0}}{m\Delta x} = 0 \tag{12}$$

Therefore, equation (11) equals to zero, which means the divergence on the fine grid is free.

V. Results and Discussion

In this section, we have applied our method to create several examples. The animation framework was implemented in C++ and rendered by PBRT. We have run all simulations including the rendering on an Intel Xeon 2.80GHz CPU with 6GB of memory.

First, we give a 2D example of smoke to demonstrate our projection method is effective, as shown in Fig. 5. Fig. 5(a),(c),(e) are the standard simulation with different resolutions and Fig. 5(b),(d) are our method reconstructed from Fig. 5(a) by scale 2 and 4, respectively. It shows that our method can generates more details than the coarse grid simulation, by comparing Figure Fig. 5(a) with Fig. 5(b),(d). It also supports that our method can produce the same level of details as the traditional high-resolution solver does, as shown in Fig. 5(b),(c) and Fig. 5(d),(e). Note that Fig. 5(f) manifests that there are little differences in detail structure between our method and traditional high-resolution grid method.

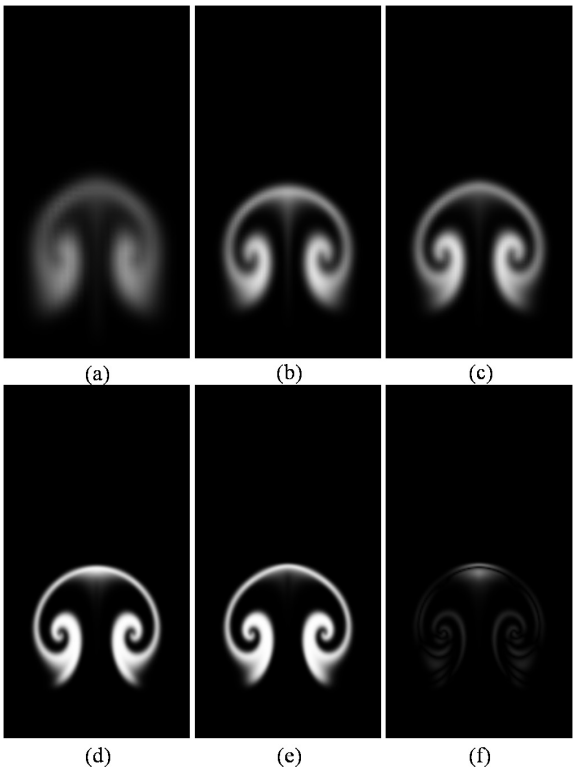

Fig. 5. 2D smoke example. (a) is a base simulation on a 32×64 grid, (b) is our method using a 64×128 grid and a 32×64 coarse grid, (c) is a base simulation on a 64×128 grid, (d) is our method using a 128×256 fine grid and a 32×64 coarse grid, (e) is a base simulation on a 128×256 grid, (f) is difference between (d) and (e).

Fig. 6 is a 2D smoke animation with an inside solid circle boundary which demonstrates that our modified coarse projection step is very effective and necessary, and it can generate more natural details than standard coarse projection. Fig. 6(b)

is the result where only using standard coarse projection, and this result is not natural and makes some artifacts. But in Fig. 6(c), using modified coarse projection, the flow passes around the object smoothly, and is more natural than Fig. 6(b).

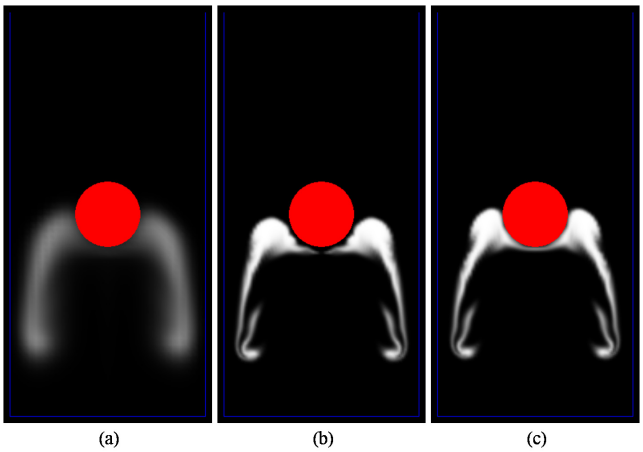

Fig. 6. 2D smoke with solid object.(a) is a base simulation on a 32×64 grid, (b) is our method using a 128×256 fine grid and a 32×64 coarse grid with standard coarse projection.(c) is our method using a 128×256 fine grid and a 32×64 coarse grid with modified coarse projection.

As in Fig. 7, Fig. 8 and Fig. 9, we show uprising 3D smoke examples interacting with both a static sphere and a moving sphere, and water in a pool. These examples are carried out with different scale of MLI-upsampling to illustrate that our method achieved much higher simulation quality than traditional solver.

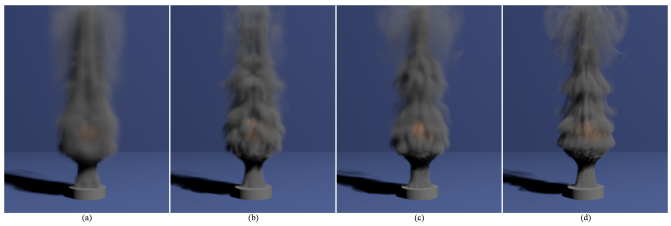

Fig. 7. 3D smoke with a moving sphere. (a) is a base simulation on a $32 \times 64 \times 32$ grid, (b) is our method using a $64 \times 128 \times 64$ grid and a $32 \times 64 \times 32$ coarse grid, (c) is base simulation on a $64 \times 128 \times 64$ grid, (d) is our method using a $128 \times 256 \times 128$ grid and a $32 \times 64 \times 32$ coarse grid.

The timings for our simulations are shown in Table 1. We compare our results by running a base simulation on the coarse grid, and a base simulation on the fine grid. The table shows our method runs approximately 152 times faster on the high-resolution in uprising smoke example.

VI. Conclusions

In this paper we introduced a novel MLI-upsampling method for fluid simulation. This upsampling method can preserve the incompressibility of velocity field when mapping from coarse grid to fine grid, without solving Poisson equation on the fine grid. Consequently, our method can provide convincing fluid details while effectively reduces the amount

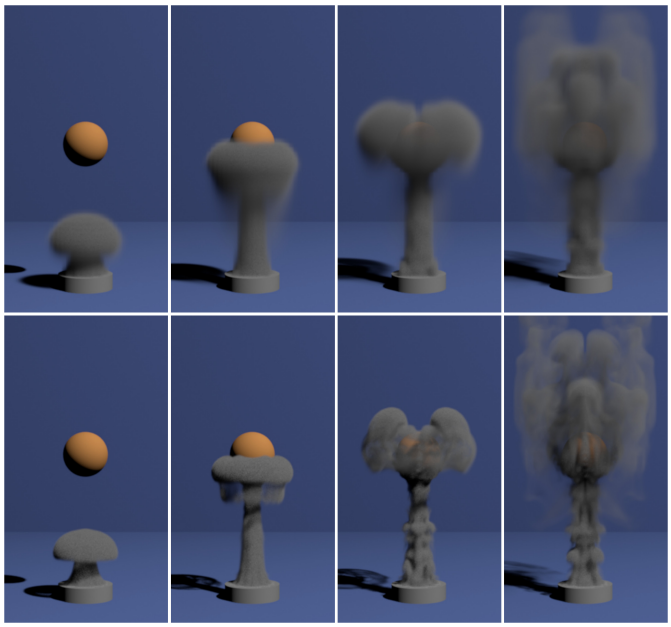

Fig. 8. 3D smoke with a static sphere. The four figures on the top is a base simulation on a $32 \times 64 \times 32$ grid with a standard simulation, while the four figures on the bottom is our method with a $128 \times 256 \times 128$ fine grid and a $32 \times 64 \times 32$ coarse grid.

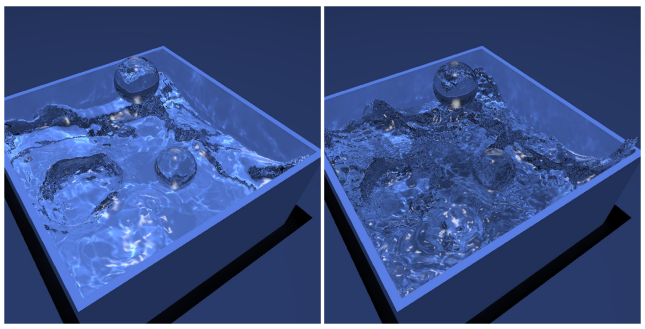

Fig. 9. water ball into a pool. left is a base simulation on a $60 \times 60 \times 60$ grid, right is our method with a $120 \times 120 \times 120$ fine grid and a $60 \times 60 \times 60$ coarse grid.

of computational time by one to two order of magnitude comparing with traditional simulator on fine grid. Besides, our technique is very simple to implement. We believe that our method is ideal for CG application where visual detail and speed are both essential.

Acknowledgment

This work was supported by a grant from the National Natural Science Foundation of China (No.60970051).

References

[1] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graph. Models and Image Processing,* vol. 58, pp. 471-483, 1996.
[2] J. Stam, "Stable fluids," in *Proc. of SIGGRAPH,* pp. 121-128, 1999.
[3] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," in *Proc. of SIGGRAPH,* pp. 15-22, 2001.
[4] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surface," *ACM Transactions on Graphics* vol. 21, pp. 736-744, 2002.

TABLE I

TIMING INFORMATION FOR OUR METHOD AND BASE SIMULATIONS ON THE FINE AND COARSE GRID. ALL TIMINGS ARE SEC-PER-FRAME.

Example	Coarse Grid	Fine Grid	Coarse	Fine	Our method	SpeedUp
2D Smoke	64×128	128×256	0.031	0.188	0.032	5.87
2D Smoke	32×64	128×256	0.015	0.188	0.016	11.75
2D Smoke	64×128	256×512	0.031	1.342	0.062	21.64
3D Smoke	$32 \times 64 \times 32$	$64 \times 128 \times 64$	1.716	45.47	2.122	21.34
3D Smoke	$32 \times 64 \times 32$	$128 \times 256 \times 128$	1.716	851.29	5.601	151.94
3D Water	$60 \times 60 \times 60$	$120 \times 120 \times 120$	4.71	254.5	5.48	46.44

[5] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Transactions on Graphics* vol. 24, pp. 965-972, 2005.

[6] T. F. Duport 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, pp. 311-324, 2003.

[7] J. Molemaker, J. M. Cohen, S. Patel, and J. Noh, "Low viscosity flow simulations for animation," in *Proc. ACM SIGGRAPH/ Eurographics Symp. Comp. Anim.*, pp. 9-18, 2008.

[8] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac, "An unconditional stable MacCormack method," *Journal of Scientific Computing* vol. 35, pp. 350-371, 2008.

[9] C. Batty, F. Bertails, and R. Bridson, "A fast variational framework for accurate solid-fluid coupling," *ACM Transactions on Graphics* vol. 26, pp. 935-941, 2007.

[10] J. Stam and E. Fiume, "Turbulent wind fields for gaseous phenomena," in *Proc. of SIGGRAPH*, pp. 369-376, 1993.

[11] N. Rasmussen, D. Nguyen, W. Geiger, and R. Fedkiw, "Smoke simulation for large scale phenomena," *ACM Transactions on Graphics* vol. 22, pp. 703-707, 2003.

[12] R. Bridson, J. Hourihan, and M. Nordenstam, "Curl-Noise for procedural fluid flow," *ACM Transactions on Graphics* vol. 26, pp. 653-655, 2007.

[13] T. Kim, N. Thurey, D. James, and M. Gross, "Wavelet turbulence for fluid simulation," in *Proc. of SIGGRAPH*, pp. 1-6, 2008.

[14] R. Narain, J. Sewall, M. Carlson, and M. C. Lin, "Fast animation of turbulence using energy transport and procedural synthesis," in *Proc. of SIGGRAPH Asia*, pp. 1-8, 2008.

[15] M. Lentine, W. Zheng, and R. Fedkiw. "A Novel Algorithm for Incompressible Flow Using Only a Coarse Grid Projection." *ACM Trans. Graph.* vol. 29(4), Article 114, 2010.

[16] R. Bridson, *Fluid Simulation For Computer Graphics*, A. K Peters, 2009.

[17] M. Muller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proc. ACM SIGGRAPH/ Eurographics Symp. Comp. Anim.*, pp. 154-159, 2003.