Pacific Graphics 2009
S. Lee, D. Lischinski, and Y. Yu
(Guest Editors)

Volume 28 (2009), Number 7

Simulating Gaseous Fluids with Low and High Speeds

Yue Gao[1] Chen-Feng Li[2] Shi-Min Hu[1] Brian A. Barsky[3]

[1] Tsinghua National Laboratory for Information Science and Technology
and Department of Computer Science and Technology, Tsinghua University
[2] Swansea University
[3] University of California, Berkeley

Abstract

Gaseous fluids may move slowly, as smoke does, or at high speed, such as occurs with explosions. High-speed gas flow is always accompanied by low-speed gas flow, which produces rich visual details in the fluid motion. Realistic visualization involves a complex dynamic flow field with both low and high speed fluid behavior. In computer graphics, algorithms to simulate gaseous fluids address either the low speed case or the high speed case, but no algorithm handles both efficiently. With the aim of providing visually pleasing results, we present a hybrid algorithm that efficiently captures the essential physics of both low- and high-speed gaseous fluids. We model the low speed gaseous fluids by a grid approach and use a particle approach for the high speed gaseous fluids. In addition, we propose a physically sound method to connect the particle model to the grid model. By exploiting complementary strengths and avoiding weaknesses of the grid and particle approaches, we produce some animation examples and analyze their computational performance to demonstrate the effectiveness of the new hybrid method.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [COMPUTER GRAPHICS]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The simulation of gaseous fluids is an important visual effect. Gaseous fluids may be slow speed, such as smoke, or high speed, such as explosions. High-speed gas flow is typically produced by severe chemical reactions, and is always accompanied by low-speed gas flow, which produces rich visual details in the fluid motion. We are interested in simulating scenes such as explosions, missile launches, and the flight of jet aircraft, etc. Realistic visualization of these dynamic phenomena involves a complex dynamic flow field with both low and high speed fluid behavior.

Eulerian grid-based methods have been shown to be effective in producing visually realistic animation for low-speed gaseous fluids. However, extending these methods to high speed gaseous fluids requires that the size of the time step be limited relative to the grid size. Empirically, [FSJ01] suggested that the time step should not exceed five times the ratio of grid size to fluid speed. Thus, to capture high speed gaseous flow in the scene, this condition requires choosing smaller time steps or larger grid size; however, so doing degrades the appearance of low speed gaseous fluids since this smears small-scale turbulence. For the simulation of compressible flow, it is noted that some recent work [KSGF09] has been done to release the stringent CFL condition imposed by the sound speed such that explicit time integration with larger time steps can be allowed in simulating inviscid compressible flow with shocks.

Due to such disadvantages of using grid approaches for modeling high-speed gaseous fluids, a Lagrangian particle method can be used instead, as was done by [TOT*03] and [RNGF03]. This approach is more suitable than the grid approaches for modeling high-speed gaseous fluids, as will be explained in Section 3. However, particle methods are not suitable for low speed gaseous fluids which can expand to a wide area, thereby necessitating a large number of samples to model small scale turbulence details which are crucial for realistic visualization. A pure Lagrangian method [TOT*03] requires a large number of particles, and finding neighboring particles becomes a system bottleneck.

Current solutions can simulate gaseous fluids addressing

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

either the low speed case or the high speed case. Although some methods extend incompressible grid-based methods to model high speed flows by using a variable density solver, these approaches make the solver complex and suffer using small time steps that are not efficient for 3D smoke simulations. The objective of this work is to develop a hybrid algorithm simulating both low speed and high speed gaseous fluids simultaneously such that visually pleasing results can be efficiently achieved. Using our approach, the low speed gaseous fluids are modeled using a grid approach to exploit the state-of-art grid-based techniques including large time step simulation and the more realistic volume rendering techniques. One the other hand, the high speed gaseous fluids are modeled by a particle approach. We also propose a physically sound transformation scheme to connect the particle model of fluids to the grid model smoothly. By using small time steps only for the high speed gaseous flow rather than for the entire simulation, our hybrid method is computationally efficient and achieves plausible results for simulating gaseous fluids with a broad range of velocity. Finally, we show some examples with low- and high- speed gaseous fluids to demonstrate the effectiveness of our hybrid method.

2. Previous Work

Jos Stam [Sta99] introduced an unconditionally stable grid-based model to the graphics community which allows large time steps to be used in fluid simulation. A fundamental assumption in his original work and many subsequent developments [FSJ01, RNGF03, SRF05, PK05, SFK*08] is that the gaseous fluid is incompressible. Incompressibility is feasible for low speed gaseous fluid simulations, but not for high speed.

Compressible fluid models have been exploited to simulate explosions. Yngve et al. [YOH00] introduced a standard explosion model where compressible Navier-Stokes equations are used to model acoustic waves and other effects. Sewall et al. [SGTL08] present an finite volume method for capturing shockwave. Compressibility complicates the algorithm and thus time steps must be small to maintain stability of the simulation. Using a grid approach, Bonner [Bon02] explored incompressible simulation method in modeling thermodynamic effects of compressible flow. Hong et al. [HSF07] integrated detonation shock dynamics to model low speed fire and flame (deflagrations) with cellular patterns. Feldman et al. [FOA03] slightly modified the Poisson equation to add artificial divergence to the velocity field. The same method is used in [IKC04] and [KJI07] to handle explosions in reactive gaseous fluids simulations. However, the divergence modification in these grid approaches is restricted and larger modifications produce unrealistic results. In fact, most detonations caused by chemical explosion are completed within microseconds [YOH00], and it is challenging for the grid-based method with divergence modification and large time steps to simulate such extreme scenarios.

Particle methods such as SPH (Smoothed Particle Hydrodynamics) have been widely used in liquid simulation [MCG03, MSKG05]. Since our work is focused on gaseous fluids, we refer to [LAD08] for more details about how SPH is used for liquid simulation. Although pure particle methods for gaseous fluids can model explosive flames [TOT*03], a large number of simulation particles are required to achieve the same realistic level as grid-based methods.

Particles are also used as a complementary tool to be combined with grid solvers. In particular, they are used to enhance flow details including turbulence, bubbles, and splashes [KCC*06, GH04, HLYK08, LTKF08] in liquid animation and sand modelling [ZB05]. For gaseous phenomena, particles are used to model particulate fuel and combustion products in explosions [FOA03], chemical reactive materials, and soot [IKC04]. Vortex particles are also widely used to increase the fluid turbulence in [AN05,PK05, SRF05]. The fundamental solvers in these hybrid works are always of one type, either grid-based or particle-based. These hybrid approaches cannot be used for our method because they cannot handle both low and high speed fluids. However, the success of this previous work gave us motivation for the development of a new hybrid approach.

3. Model and Algorithm

The actual fluid with broad range of velocity is a whole system where low- and high- speed fluids are inherently mixed. For grid-based methods, small time steps have to be used globally in order to resolve the high-speed fluid components. For particle-based methods, a large amount of particles are required to sample the turbulent low-speed fluid components. In this challenging scenario, it is difficult to solve efficiently the mixed fluid through grid-based or particle-based methods alone because of their discretization limits.

As shown in Figure 1, we propose to divide the fluid into low- and high- speed parts, which are simplified and solved by particle and grid approaches respectively, and then synthesize the final gaseous density fields for rendering. The particle and grid solvers are connected through mutual drag forces [FOA03, KJI07] and a fluid conversion mechanism where the fluid represented by particles is automatically transformed into a grid representation when its velocity falls below a threshold.

The high-speed fluid component is modeled by an explicit SPH method. Its advantages are threefold [LLZL03]:

1. The lifetime of high speed gas is very short. For example in the case of explosions, chemical detonation is completed within microseconds. Explicit integral schemes are therefore more suitable in this case.
2. High speed gas emitting from a small source into a large open space leads to a significant change of density fields. Particles approaches inherently conserve fluid mass locally within the particle during advection.
3. High speed gas flow often happens with drastic volume expansion, and subsequently the velocity field is no

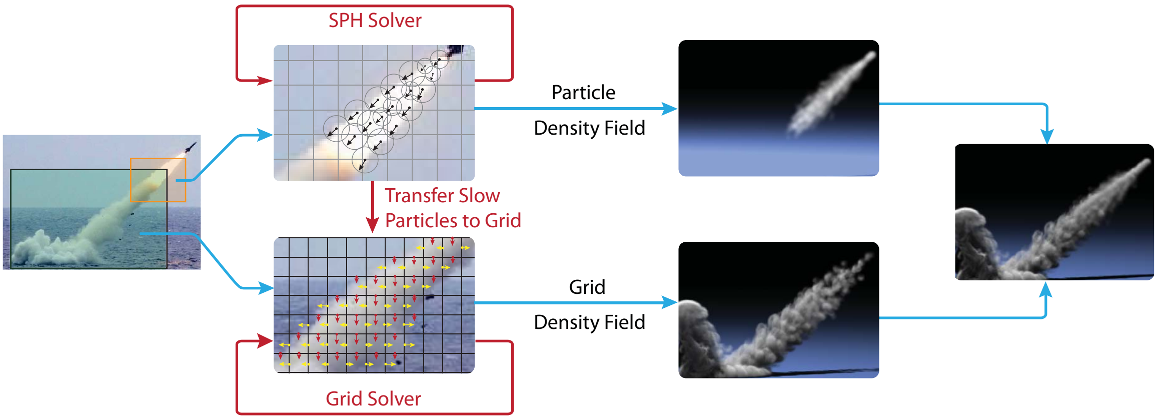

Figure 1: *The hybrid particle-grid model for gaseous fluids with low and high speeds.*

longer divergence-free. Particle approaches such as SPH have the compressible problem in solving divergence-free fields (weakly compressible caused by the numerical discretization format), but we found that its "weakness" visually mimics gaseous expansion better and is also more robust than the divergence modification scheme used in grid approaches.

The low speed fluid component is modeled by an implicit grid method because incompressibility and sufficient sampling for small scale turbulence details become crucial for realistic visualization. Also, using a grid model for low-speed gas makes it easier to exploit state-of-art grid-based techniques including simulation and more realistic volume rendering techniques.

It should be noted that the fluid as a whole is governed by the Navier-Stokes equation and due to its nonlinearity, splitting the system into two parts satisfying separate N-S equations is not mathematically rigorous and cannot provide an exact solution for the original system. However, our contributions are in the technical aspect rather than in the fluid dynamics theory. Like various simulation algorithms used in graphics, the plausible visual effects are the major concerns here. Mass and momentum conservation plays the dominant role in determining the evolution of gaseous fluid and its visual effects. In the proposed hybrid model, mass and momentum conservation is retained separately for the high speed and low speed fluid components. Also, when converting gaseous fluid from the particle representation to the grid model, an effective pressure is computed according to the ideal gas law and this enables the visual effects of mass and momentum conservation to be captured during the transformation. Separation of low speed and high speed fluid components allows the governing N-S equation to be simplified in accordance with different physical nature of fluids and then solved efficiently with specialized solvers. The examples we made shows that the hybrid model maintains good simulation efficiency for the whole system and produces virtually pleasing results.

Specifically, both the particle and grid solvers are fundamentally based on the N-S equations that conserve mass and momentum:

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{u} \tag{1}$$

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{f} \tag{2}$$

where $\mathbf{u}$ is the velocity, p the pressure, ρ the density of the fluid, μ the viscosity and $\mathbf{f}$ any external force including buoyancy, vorticity confinement, drag force or reaction force between particles and grids etc. To simplify the system and reduce its nonlinearity, physically rigorous equations for energy conservation and state change are not included in the hybrid model.

3.1. Smoothed Particle Hydrodynamics (SPH)

The SPH method, a pure Lagrangian solver, is used to simulate high-speed gas in Figure 1. The SPH particle i is accelerated according to the sum of forces exerted by adjacent particles, $\mathbf{f}_{ij}$, as

$$\frac{d\mathbf{v}_i}{dt} = a_i = \frac{\sum_j \mathbf{f}_{ij}}{\rho_i} \tag{3}$$

where $\rho_i = \sum mW(\mathbf{x}_{i,j}, r_i)$ is the density of particle i and we unify the mass m for all particles. To avoid confusion, we use $\mathbf{v}$ to denote particle velocity and $\mathbf{u}$ for grid velocity. The velocity and position of each particle can be computed by standard ODE integration techniques; the leap-frog Euler approach is used here.

The same approach as [APKG07] is employed to determine the pressure and viscosity forces,

$$\mathbf{f}_{ij}^{pressure} = -V_iV_j\frac{P_i+P_j}{2}(\nabla W(\mathbf{x}_{ij}, r_i) + \nabla W(\mathbf{x}_{ij}, r_j)) \tag{4}$$

$$\mathbf{f}_{ij}^{viscosity} = \mu V_iV_j\frac{\mathbf{v}_j-\mathbf{v}_i}{2}(\nabla^2 W(\mathbf{x}_{ij}, r_i) + \nabla^2 W(\mathbf{x}_{ij}, r_j)) \tag{5}$$

where $V_i = m/\rho_i$ is the particle volume, $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ and

$W(\mathbf{x}, r)$ the radially symmetric kernel function with support r. We use the kernel functions defined in [MCG03] for each kind of SPH approximation. In contrast to the constant kernel radius used in liquid fluid simulations [MCG03, BT07], the kernel radius is variable to take advantage of adaptive resolution of SPH for gas simulation:

$$r_i = k_r \left(\frac{\rho_i}{\rho_{i0}} \right)^{-\frac{1}{3}} r_0 \tag{6}$$

where ρ_{i0} is the SPH density before the simulation starts, r_0 the initial radius (we use the same value for all particles), and k_r is a scaling parameter set to 1 in our implementation. An upper bound of r_i is needed for stability purposes; we set it to $2r_0$.

In addition, the state equation of ideal gas is included to connect the pressure P_i and density ρ_i, as follows

$$P_i = k_p(\rho_i - \rho_0) \tag{7}$$

with pressure constant k_p. ρ_0 is a tuning parameter to provide numerical stability [MCG03]. A drag force $\mathbf{f}$ is also exerted on each particle:

$$\mathbf{f} = k_d(\bar{\mathbf{u}} - \mathbf{v}) \parallel \bar{\mathbf{u}} - \mathbf{v} \parallel \tag{8}$$

where k_d is the drag force coefficient controlling the interaction between grids and high speed particles, $\bar{\mathbf{u}}$ is the grid's interpolated velocity at the particle location. The opposite drag force is also exerted on the grid cell containing the particle.

3.2. Grid

The low-speed gas flow in Figure 1 is modeled as an incompressible inviscid fluid. Thus the viscosity term $\mu\nabla^2\mathbf{u}$ in Eqn.2 is ignored, and the mass conservation condition Eqn.1 becomes,

$$\nabla \cdot \mathbf{u} = 0 \tag{9}$$

Buoyancy, vorticity confinement and reaction drag force comprise the external forces $\mathbf{f}$ in Eqn. 2, and the grid temperature evolves following

$$\frac{\partial T}{\partial t} = -(\mathbf{u} \cdot \nabla)T + k_T \nabla^2 T \tag{10}$$

with diffusion coefficient k_T. Numerical methods for solving the above equations have been well studied and readers are referred to [Sta99, FSJ01, BMF07] for more details.

3.3. Transfer Particles to Grid

As gas expands in space, the velocity decreases rapidly so that using the compressible flow model and dense particle sampling are no longer feasible. Therefore, to maintain an efficient simulation, particle quantities including fluid mass and velocity, smoke density and particle temperature must be mapped onto grids, and the simulation continued with a grid-based method.

The conversion is based on a velocity criterion. Particles whose velocity falls below a threshold are transferred to grids and eliminated from the particle solver. The threshold is a tuning parameter; in all examples we set it to 5 times the grid cell length divided by the time step used in the grid solver. It is observed in numerical experiments that a few local grid cells may occasionally be accelerated such that their velocities temporarily rise above the velocity threshold. This occurs because the SPH particles nearby are being converted to the grid model. However, the transformation from the grid model to the particle model is not considered here because the conversion from particles to grids dominates.

Physical quantities of a particle i with mass m and velocity v_i is transferred to neighboring grid cells within a user-defined transferring radius r_t. First, the particle fluid mass m is formally distributed among neighboring cells as

$$m_j = \frac{mG(d_{ij}, r_t)}{\sum_{k \in C_i} G(d_{ik}, r_t)} \tag{11}$$

where $G(d_{ij}, r_t)$ is a Gaussian weight kernel with r_t as the support radius, d_{ij} denotes the distance between particle i and the center of cell j and C_i is the set of neighboring cells for particle i. Then, according to momentum conservation, the fluid velocity in neighboring cells is increased by

$$\Delta\mathbf{u}_j = \frac{m_j(\mathbf{v}_i - \mathbf{u}_j)}{m_g + m_j} \tag{12}$$

where m_g is the constant cell fluid mass in the grid solver and $\mathbf{u}_j$ the fluid velocity in cell j before velocity updating. To satisfy mass conservation, mass m_j should contribute to the total fluid mass in cell j in the grid model. However, as fluid density is a constant in the incompressible fluid model, the additional mass contribution can not be directly included in the grid solver. According to the ideal gas law $PV = nRT$, where the pressure P is proportional to the mass term n when temperature T and volume V are fixed, we introduce an effective pressure to capture the effect of mass conservation during the transformation. Specifically, the effective pressure generated by the additional mass m_j in cell j is

$$\Delta p_j = \beta m_j, \tag{13}$$

where β is a tuning factor depending on the grid size. The effective pressure Δp_j is then directly added onto the pressure field obtained in the grid solver. The effective pressure indirectly introduces divergence to grid velocity field within the transformation region. The whole effective pressure field is smooth, as it is the sum of each Gaussian filtered particle effective pressure. Figure 4 shows that this approach is more robust and able to generate more realistic results than the artificial divergence modification [FOA03].

In the SPH model, particle smoke density is $\rho^s = \alpha m$ where α is a tuning factor to adjust the amount of smoke density carried by each simulation particle. Thus, redistribution of particle fluid mass m in Eqn. 11 will also cause an increase of smoke density in the neighboring cells as

$$\Delta\rho^s_j = \alpha m_j. \tag{14}$$

The artificial temperature is updated in the same way as velocity

$$\Delta T_j = \frac{m_j(t_i - T_j)}{m_g + m_j}, \tag{15}$$

where t_i is the user-defined particle temperature of particle i and T_j the user-defined grid temperature of cell j before temperature updating.

These mapping schemes will generate various initial conditions for the grid solver in terms of velocity and pressure of fluid, smoke density and grid temperature. Throughout the particle-grid transformation, the effect of mass and momentum conservation is fully taken into account. However, when only a few particles are transferred to grids, undesirable discontinuous density artifacts may appear, especially in the initial stage of simulation; increasing the number of particles resolves this issue. Since particle transfer always takes place in a relatively small space in the whole simulation domain, a few hundred to a few thousand of particles are usually sufficient to suppress the artifacts.

3.4. Algorithm

Based on above model, we have our hybrid particle-grid method as follows:

1. Initialize source particle positions and velocities.
2. Particle phase, for each particle time step:
   - Compute SPH density ρ_i and radius r_i.
   - Compute the net force on each particle.
   - Update particle velocities and positions according to the external forces due to particle interaction force and the drag force from grids to particles.
   - For low-speed particles, transfer fluid mass and velocity, smoke density and particle temperature to grids, and eliminate them from the particle system.
   - For high-speed particles, transfer density to a temporary grid just for rendering.
3. Grid phase:
   - Perform the force addition, advection steps.
   - Compute the initial pressure from the classic projection poisson equation.
   - Add the effective pressure that accounts for the increment of grid mass to the pressure field.
   - Subtract the gradient of total pressure from the grid velocity field.
4. Output the sum of density fields in the grid solver and the temporary grid, then clear the temporary grid.
5. Go back to step 2.

The particle time step is smaller than the grid time step. In our implementation, the time step of the grid solver is set to animation frame rate, such as 1/30 second, and the time step of the particle solver varies from 1/10 to 1/3 of the grid time step depending on initial particle velocity. Detection of neighboring particles is accelerated with the help of a background mesh, so the simulation time spent in each particle step is significantly reduced.

Scene	Grid Size	SPH solver	Grid solver
Missile(Fig.2)	192x128x128	0.36s	30.78s
Jet(Fig.3 Row 4)	128x128x128	3.03s	17.17s
Arch(Fig.4 Row 2)	64x64x64	0.03s	3.18s
Rocket(Fig.5)	128x128x128	4.83s	27.52s

Table 1: *Performance of the hybrid method. Listed in the right two columns are the average time cost to generate one frame.*

Parameter	Missile	Jet	Arch	Rocket
dx	0.0052	0.0078	0.0156	0.0078
m	0.03	0.05	0.07	0.05
r_0	0.020	0.015	0.015	0.015
k_p	0.7	1.0	0.6	0.4
k_d	8.5	3.2	5.0	4.5
ν	80	80	20	80
r_t	0.012	0.015	0.009	0.015
α	0.8	0.6	0.4	0.8
β	0.12	0.2	0.32	0.2

Table 2: *Simulation parameters.*

In some cases like a rocket launch or aircraft jet engine, step 1 has to be moved into the particle phase as a persistent fluid source.

4. Results

The hybrid particle-grid method has been implemented in C++, and all examples were tested on a PC with a 2.7GHz Core2 Duo Intel CPU and 2GB memory. The animation speed is 30 frames per second and all images were rendered by pbrt [PH04]. Four examples are presented in this section, for which the performance and simulation parameters are listed in Tables 1 and 2 respectively. It can be seen in Table 1 that the cost of the particle solver is very moderate because the new hybrid approach only requires a small number of SPH particles. However, the following examples show that for modeling gaseous fluid with low and high speeds, the small extra cost paid for use of particles significantly improves the performance of conventional grid-based methods in computer graphics.

Figure 2 shows an example of a missile launch. Constant high-speed gas is generated from the exhaust at the bottom of the missile and it drives the surrounding low-speed gas to produce rich dynamic visual details. Thirty particles are added as gas source every particle time step. When the simulation reaches a balancing state, i.e. when the number of newborn SPH particles being generated from the missile exhaust equals the number of dying SPH particles being mapped into the grid, there are circa 350 particles in a single frame. By combining strengths of particle and grid methods, the new hybrid approach allows complex gaseous flow moving with low and high speeds to be modeled with moderate computational cost. Although the low- and high- speed fluids are simulated separately, smooth and seamless visual

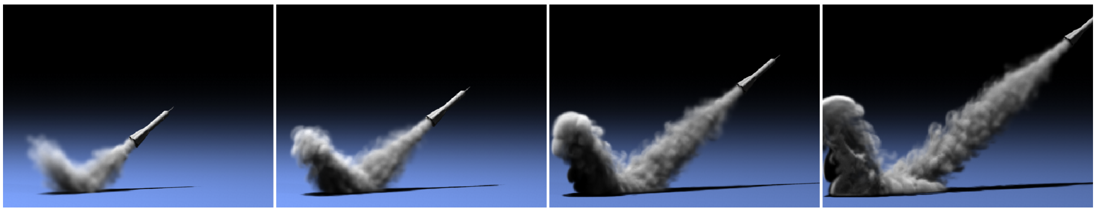

Figure 2: *Missile launch. Represented by SPH (Smoothed Particle Hydrodynamics) particles, high speed gas near the missile exhaust drives the fluids. Simulated by the grid solver, low-speed gas at far field produces rich visual details.*

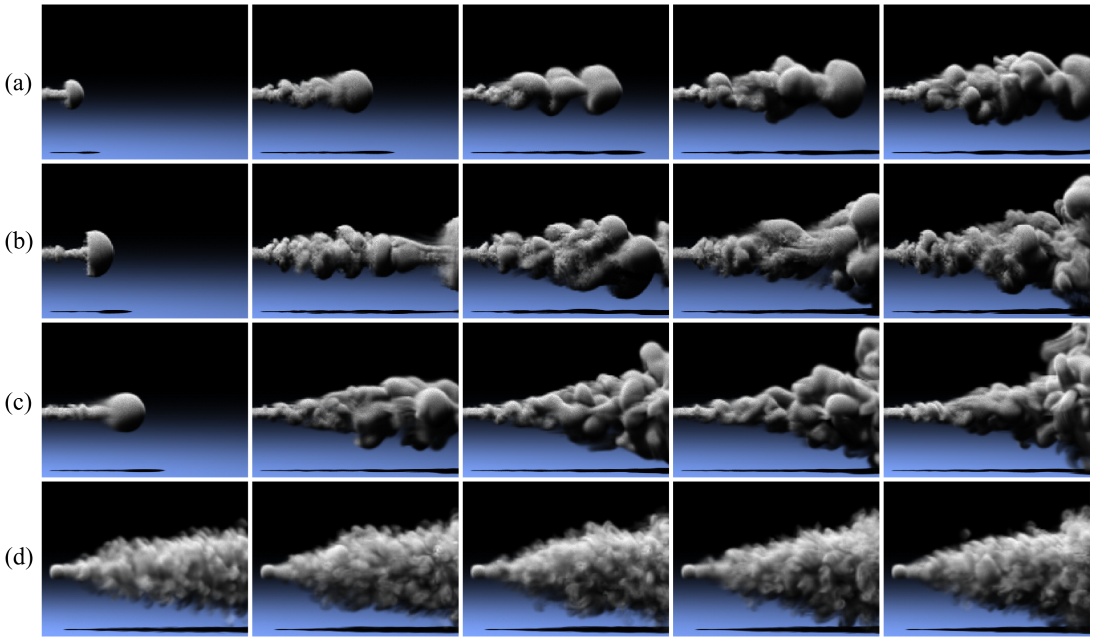

Figure 3: *Jet. From left to right, images are taken every 15 frames with the first column starting from the 10th frame. (a), (b) and (c) are obtained with the traditional grid-based solver. (d) is obtained with the proposed hybrid method. From (a) to (d), (Initial speed of the source gas, Time step in the grid solver) are set respectively as (150, 1/30), (450, 1/30), (450, 1/90) and (450, 1/30).*

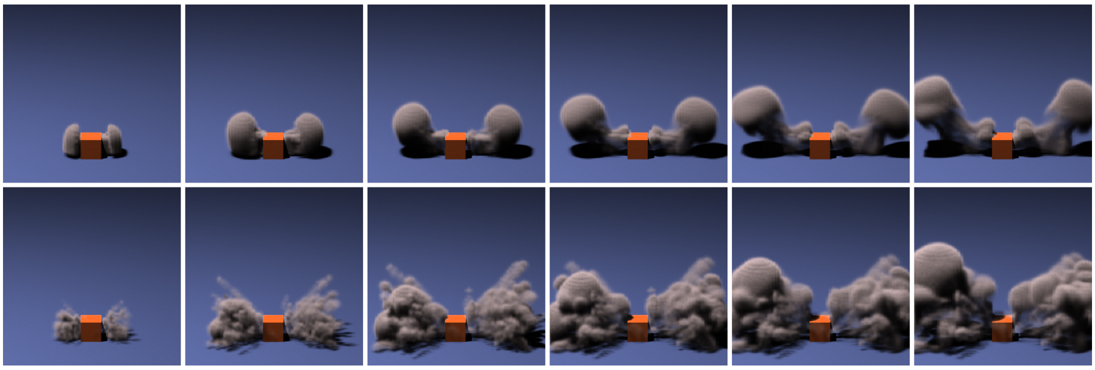

Figure 4: *Arch. From left to right, images are taken every 9 frames with the first column starting from the 6th frame. The top row is obtained by using the grid-based approach combined with artificial divergence modification. The bottom row is obtained by using the proposed hybrid approach in which 800 SPH particles are used.*

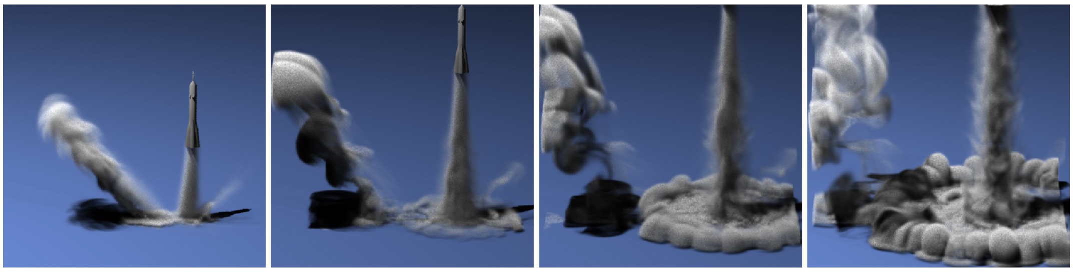

Figure 5: *Rocket Launch. As that in reality, a V-shaped underground channel is built below the rocket to facilitate the transport of explosive gas generated during rocket launch.*

effects are achieved by the automatic particle-grid conversion scheme that ensures the effects of mass and momentum conservation between the two solvers.

Figure 3 shows a jet example, and for different speeds of source gas, the new hybrid method is compared with the conventional grid-based method. In row (a) where the speed of source gas is 5 times the ratio of cell length to time step, the grid-based method is observed to give physically rational results but with limited details for laminar and turbulent motions. In row (b) where the speed of source gas is further increased by a factor of 3, the result obtained by the grid method becomes physically unreasonable although the simulation is still stable. In row (c) where the time step is reduced by a factor of 3 to compensate for the increased speed from the gas source, the grid method can correctly capture the laminar motion but with a significant loss of turbulence detail. In row (d), for the same speed gas source as (c), and without using smaller time steps, laminar and turbulent motions are both better captured by the new hybrid method. It can be seen in Figure 3 (d) that at the larger scale the jet flow evolves like a laminar fluid and at the smaller scale, the flow is accompanied by rich turbulence detail. Forty particles are added as gas source at every particle time step and there are about 600 particles in the frame when balance is achieved.

An arch example is shown in Figure 4 where the new hybrid method is compared with a grid-based method containing artificial divergence modification. Results obtained by the grid approach are given in the top row. As the source generating non-zero divergence is placed at the center, the induced velocity field is symmetric in space and correspondingly its visual effects appear to be flat especially in the first serval frames. The numerical experiments show that the divergence value must be set carefully in order to avoid discontinuity artifacts in the density field. In this example, the divergence is set as 0.01 and placed across a 3x3 grid at the center of the scene; to generate a gaseous flow with free expansion, it is only switched on for half a second at the beginning of the simulation. The result obtained by the new hybrid method is shown in the bottom row. A total number of 800 static SPH particles are placed at the center of the arch, and the internal pressure between SPH particles triggers an explosion effect followed by free expansion. All particles are transferred to the grid within 8 frames. If particles collide with obstacles, they are moved back along the normal direction in the next time step. Controlled by the automatic particle to grid transformation, these particle-particle and particle-obstacle interactions produce various scales of visual details for the low-speed gaseous fluid. The result shows that many visual details are produced from the beginning of the simulation and are preserved through the animation.

Figure 5 shows a rocket launch. At the beginning of the launch, smoke travels through the underground channel, and then sprays out from the side exit. As the rocket rises, more and more smoke accumulates and spreads on the ground. The particle method can easily handle these complex boundaries, especially for the high-speed gas. In this example, 50 particles are injected every time step, and there are about 520 particles per frame when balance is reached.

5. Conclusion

Simulating and rendering fluids with low and high speeds is a challenging task, and currently neither grid-based nor particle-based methods can provide satisfactory solutions alone because of their discretization limits. We present a hybrid method for simulating gaseous fluids with low and high speeds, in which the fluid is formally split into high velocity and low velocity parts. Although the two fluid components share the same domain in space, they are simulated separately using particle and grid approaches, after which the final gaseous density fields are synthesized for rendering. The interaction between high-speed and low-speed fluids is modeled by mutual drag forces in the particle and grid solvers, and the high-speed gaseous fluid is also automatically transformed from the particle representation to the grid model when its velocity falls below a threshold. A particle to grid conversion scheme that ensures the effects of mass and momentum conservation is used to automatically transform the particle-represented fluid into a grid representation, so that both the particle and the grid solvers are put to best use to solve the complex dynamic flow with minimum computational cost. By exploiting the complementary strengths and avoiding weaknesses of grid and particle methods, we can animate such complex scenes as missile launch, explosions, flight of jet aircraft, etc. where low- and high- speed flow both play important roles.

The limitation of this hybrid model is in representing the accurate physics. First, due to the nonlinearity of the Navier-Stokes equation, splitting the fluid into two components that satisfy separate N-S equations is not mathematically rigorous. Secondly, although chemical reactions and phase change are always involved in phenomena such as explosions and missile launch in the real world, governing equations of energy conservation and state change are not included in the hybrid model in order to simplify the nonlinearity of the system. Thirdly, discontinuous effects such as shock wave cannot be modeled by our method. However, like many simulation algorithms in graphics, producing visually pleasing results with moderate computational cost is the major concern in this work and this goal has been achieved as demonstrated in the examples. The success of the hybrid algorithm owes to its efficient and correct capture of the effects of mass and momentum conservation, which is the dominant physics for evolution and corresponding visual effects of gaseous fluids.

6. Acknowledgement

This work was supported by the National Basic Research Project of China (2006CB303105), the Natural Science Foundation of China (U0735001) and the National High Technology Research and Development Program of China (2009AA01Z330)

References

[AN05] Angelidis A., Neyret F.: Simulation of smoke based on vortex filament primitives. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), pp. 87–96.

[APKG07] Adams B., Pauly M., Keiser R., Guibas L. J.: Adaptively sampled particle fluids. In *ACM SIGGRAPH* (2007), p. 48.

[BMF07] Bridson R., Müller-Fischer M.: Fluid simulation. In *ACM SIGGRAPH courses* (2007), pp. 1–81.

[Bon02] Bonner M.: *Compressible Subsonic Flow on a Staggered Grid*. Master's thesis, University of British Columbia, 2002.

[BT07] Becker M., Teschner M.: Weakly compressible sph for free surface flows. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 209–217.

[FOA03] Feldman B. E., O'Brien J. F., Arikan O.: Animating suspended particle explosions. In *ACM SIGGRAPH* (2003), pp. 708–715.

[FSJ01] Fedkiw R., Stam J., Jensen H. W.: Visual simulation of smoke. In *ACM SIGGRAPH* (2001), pp. 15–22.

[GH04] Greenwood S. T., House D. H.: Better with bubbles: enhancing the visual realism of simulated fluid. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 287–296.

[HLYK08] Hong J.-M., Lee H.-Y., Yoon J.-C., Kim C.-H.: Bubbles alive. In *ACM SIGGRAPH* (2008), pp. 1–4.

[HSF07] Hong J.-M., Shinar T., Fedkiw R.: Wrinkled flames and cellular patterns. In *ACM SIGGRAPH* (2007), p. 47.

[IKC04] Ihm I., Kang B., Cha D.: Animation of reactive gaseous fluids through chemical kinetics. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 203–212.

[KCC*06] Kim J., Cha D., Chang B., Koo B., Ihm I.: Practical animation of turbulent splashing water. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), pp. 335–344.

[KJI07] Kang B., Jang Y., Ihm I.: Animation of chemically reactive fluids using a hybrid simulation method. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 199–208.

[KSGF09] Kwatra N., Su J., Grétarsson J. T., Fedkiw R.: A method for avoiding the acoustic time step restriction in compressible flow. *J. Comput. Phys. 228*, 11 (2009), 4146–4161.

[LAD08] Lenaerts T., Adams B., Dutré P.: Porous flow in particle-based fluid simulations. In *ACM SIGGRAPH* (2008), pp. 1–8.

[LLZL03] Liu M. B., Liu G. R., Zong Z., Lam K. Y.: Computer simulation of high explosive explosion using smoothed particle hydrodynamics methodology. *Computers & Fluids 32*, 3 (2003), 305 – 322.

[LTKF08] Losasso F., Talton J., Kwatra N., Fedkiw R.: Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (July-Aug. 2008), 797–804.

[MCG03] Müller M., Charypar D., Gross M.: Particle-based fluid simulation for interactive applications. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), pp. 154–159.

[MSKG05] Müller M., Solenthaler B., Keiser R., Gross M.: Particle-based fluid-fluid interaction. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), pp. 237–244.

[PH04] Pharr M., Humphreys G.: *Physically Based Rendering : From Theory to Implementation*. August 2004.

[PK05] Park S. I., Kim M. J.: Vortex fluid for gaseous phenomena. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), pp. 261–270.

[RNGF03] Rasmussen N., Nguyen D. Q., Geiger W., Fedkiw R.: Smoke simulation for large scale phenomena. In *ACM SIGGRAPH* (2003), pp. 703–707.

[SFK*08] Selle A., Fedkiw R., Kim B., Liu Y., Rossignac J.: An unconditionally stable maccormack method. *J. Sci. Comput. 35*, 2-3 (2008), 350–371.

[SGTL08] Sewall J., Galoppo N., Tsankov G., Lin M.: Visual simulation of shockwaves. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), pp. 19–28.

[SRF05] Selle A., Rasmussen N., Fedkiw R.: A vortex particle method for smoke, water and explosions. In *ACM SIGGRAPH* (2005), pp. 910–914.

[Sta99] Stam J.: Stable fluids. In *ACM SIGGRAPH* (1999), pp. 121–128.

[TOT*03] Takeshita D., Ota S., Tamura M., Fujimoto T., Muraoka K., Chiba N.: Particle-based visual simulation of explosive flames. In *The 11th Pacific Conference on Computer Graphics and Applications* (2003), p. 482.

[YOH00] Yngve G. D., O'Brien J. F., Hodgins J. K.: Animating explosions. In *ACM SIGGRAPH* (2000), pp. 29–36.

[ZB05] Zhu Y., Bridson R.: Animating sand as a fluid. In *ACM SIGGRAPH* (2005), pp. 965–972.