

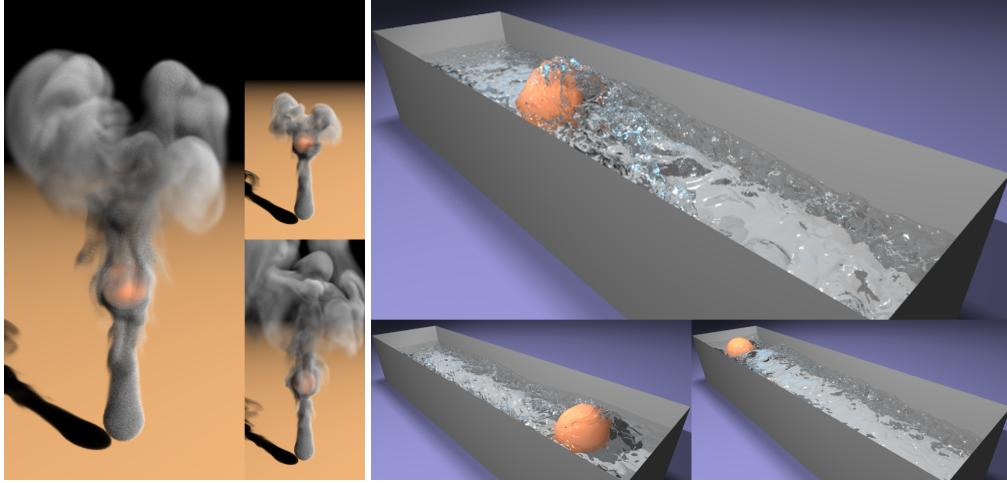
# Fast Animation of Turbulence Using Energy Transport and Procedural Synthesis

Rahul Narain\*  
UNC Chapel Hill

Jason Sewall†  
UNC Chapel Hill

Mark Carlson‡  
Dreamworks Animation SKG

Ming C. Lin§  
UNC Chapel Hill



**Figure 1:** Smoke swirls around an obstacle (left); and a moving ball creates a turbulent wake behind it (right).

## Abstract

We present a novel technique for the animation of turbulent fluids by coupling a procedural turbulence model with a numerical fluid solver to introduce subgrid-scale flow detail. From the large-scale flow simulated by the solver, we model the production and behavior of turbulent energy using a physically motivated energy model. This energy distribution is used to synthesize an incompressible turbulent velocity field, whose features show plausible temporal behavior through a novel Lagrangian approach for advected noise. The synthesized turbulent flow has a dynamical effect on the large-scale flow, and produces visually plausible detailed features on both gaseous and free-surface liquid flows. Our method is an order of magnitude faster than full numerical simulation of equivalent resolution, and requires no manual direction.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

## 1 Introduction

Visual simulation of fluids provides a powerful tool for realistic animation of many complex phenomena, such as smoke and water. However, to obtain visually appealing fine details in the large-scale

turbulent fluids, a mesh with very high spatial resolution is often required for capturing the details, thus the associated computational cost can be prohibitively high. As an alternative, Procedural techniques for animating fluids can represent irregular turbulent flow, but require careful animator control and are not always applicable to completely general flow scenarios.

We present a novel method for augmenting numerical fluid simulation with a stochastic procedural model for subgrid-scale details to achieve fast visual simulation of turbulent fluids. Our turbulence model automatically estimates the intensity of small-scale details induced by the large-scale flow, and introduces swirling eddies and vortices into the fluid using a fast procedural technique inspired by the Kolmogorov law for turbulent flows [Frisch 1995]. Our approach has the following characteristics:

- It both compensates for the loss of fine-scale details due to numerical dissipation and introduces additional subgrid-scale flow which cannot be captured on the simulation grid at all.
- It accounts for the two-way interaction of the procedurally synthesized turbulence with the large-scale simulated flow, enabling the synthesized turbulent flow to affect the dynamics of the numerical simulation for added realism.
- There is only a loose coupling between the fluid solver and the procedural turbulence model, which allows the two to run at different resolutions independently.
- It is general and applicable to a wide spectrum of fluids, including swirling gaseous flows and complex river rapids with free surfaces.

Our approach can yield very small-scale fluid details from procedural synthesis, while requiring only a low-cost fluid solver on a coarse mesh and achieving an order of magnitude performance gain over high-resolution fluid simulation that produces comparable details. Fig. 1 shows an animation sequence of roiling smoke around an obstacle (left) and the turbulent wake created by a moving ball in a tank of water (right) synthesized by our method. Furthermore, our technique also allows for flexibility in the choice of the numerical fluid solvers.

\*e-mail: narain@cs.unc.edu

†e-mail: sewall@cs.unc.edu

‡e-mail: mark.t.carlson@gmail.com

§e-mail: lin@cs.unc.edu

## 2 Previous Work

The study of turbulence has a long history in the fluid dynamics literature, and remains one of the last open problems in classical mechanics. Seminal work was done in this area by Richardson, Kolmogorov, and Taylor in the early 20th century. The field has seen an enormous body of work since then, but many results remain qualitative, semi-empirical, or restricted to special cases, and a general theory of turbulence remains elusive. Due to the stochastic nature of turbulent fluctuations, a statistical approach has proved useful in this domain (see [Monin and Yaglom 1971; McComb 1990]). For nonspecialists, an accessible introduction to this challenging field of physics can be found in [Davidson 2004].

In computer graphics, physically based animation of fluids such as smoke and water has received considerable attention in recent years. Numerous techniques have been proposed using both Eulerian [Foster and Metaxas 1996; Stam 1999; Foster and Fedkiw 2001] and Lagrangian approaches [Müller et al. 2003; Premoze et al. 2003]. Some approaches work with vorticity rather than velocity [Angelidis and Neyret 2005; Park and Kim 2005; Elcott et al. 2007]. We refer the reader to recent papers [Chentanez et al. 2007; Adams et al. 2007] for a more thorough review.

One of the prominent difficulties faced in fluid simulation, especially in the popular Eulerian approach, is loss of small eddies and vortices due to numerical diffusion. Consequently, much work has been directed towards avoiding or counteracting this phenomenon to achieve a more detailed, lively appearance in the fluid. Fedkiw et al. [2001] proposed vorticity confinement to amplify small-scale vortices in the fluid. However, if the simulation grid is not fine enough to capture the desired details, vorticity confinement cannot recover them. Selle et al. [2005] discuss this limitation and propose vorticity particles to introduce additional vorticity into the fluid for highly turbulent flows. Their method requires the artist to specify where these particles are injected into the flow, and does not reproduce the decay of turbulence when the forcing is removed; thus, it is best suited for scenarios where a constant source of turbulent vorticity is known *a priori*. Another approach is to use an error correction scheme such as BFECC [Kim et al. 2007] or a MacCormack method [Selle et al. 2008], or a less dissipative advection scheme such as USCIP [Kim et al. 2008a], to reduce the diffusion in the numerical method directly. All of these above methods work directly on the simulation grid itself, thus none of them can introduce subgrid-scale vortices into the fluid. (We note that in the context of particle-based fluid simulation, the FLIP method [Brackbill and Ruppel 1986; Zhu and Bridson 2005] shows little to no numerical dissipation.)

Procedural methods for modeling fluid flow are often used by practitioners since they offer cheap evaluation and controllability. Early work synthesized flow fields using a superposition of primitives for laminar flow [Sims 1990; Wejchert and Haumann 1991], while Fourier synthesis for turbulent flow [Shinya and Fournier 1992; Stam and Fiume 1993] did not allow for spatial modulation of the intensity of turbulence. Recently, curl noise [Kniss and Hart 2004; Bridson et al. 2007] has been introduced, allowing spatially varying incompressible flow fields that respect rigidly moving boundaries. While curl noise can be used to synthesize a turbulent flow field by adding several octaves of noise, it requires the spatial modulation of turbulence to be specified. We fill this gap by automatically determining the distribution of turbulent energy corresponding to a complex large-scale flow.

In concurrent work to ours, Kim et al. [2008b] used wavelet analysis to determine the characteristics of missing turbulent flow components and synthesize them with band-limited wavelet noise. While their motivation is similar to ours, the local wavelet-based

approach is a novel technique which is very different from our idea of tracking the dynamics of turbulent energy over time. Much closer to our approach is another concurrent work by Schechter and Bridson [2008], which tracks several bands of turbulent energy using a simple linear model and generates the turbulent velocity using flow noise [Perlin and Neyret 2001]. Another novel contribution is that they analyzed and corrected the additional vorticity dissipation due to time splitting of the pressure and advection. Their approach is conceptually easy to implement, but requires an animator to seed a distribution of turbulent energy. We describe a way of automatically generating a reasonable distribution using the large-scale flow, reducing the amount of manual labor.

As compared to the concurrent work [Kim et al. 2008b; Schechter and Bridson 2008], one of the significant differences is that we consider the application of procedural turbulence to animation of liquids with free surfaces, while the concurrent work has mainly been applied to only smoke simulation. We will discuss some of the challenges in coupling procedural turbulence with liquids in section 5.3. In addition, we keep some overlap between the scales of synthesized turbulence and those of the large-scale flow, allowing for the turbulence to introduce fluctuations in the large-scale flow as well for a more visually pleasing flow. Concurrent works keep the turbulence independent from the large-scale flow, so that turbulence can then be added as a post-process for ease of editing. Our method can support this feature if desired simply by running it on strictly smaller scales than the grid resolution with the backwards coupling disabled.

## 3 Statistical Modeling of Turbulence

The behavior of a viscous, incompressible fluid is described by the Navier-Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla^2 p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

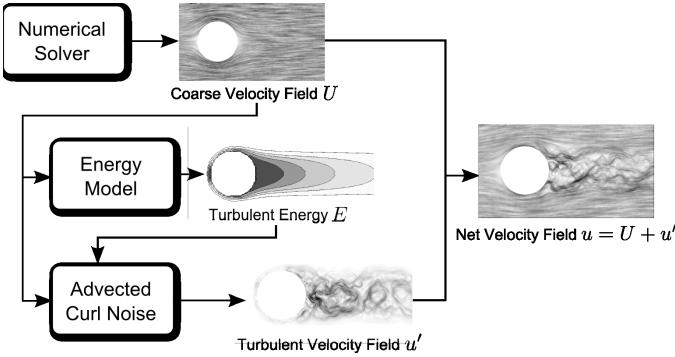
where  $\mathbf{u}$  is the fluid velocity,  $p$  is pressure,  $\nu$  is the kinematic viscosity of the fluid,  $\mathbf{f}$  is the external force, and  $\nabla = [\partial/\partial x, \partial/\partial y, \partial/\partial z]^T$ . The density of the fluid is taken to be unity.

Fluids obeying the Navier-Stokes equations show smooth laminar flow at low speeds, but as the speed is increased or the viscosity lowered, turbulence arises and the flow becomes chaotic, with irregular fluctuations at large and small scales. Many fluid phenomena of visual interest exhibit a high degree of turbulence, and the range of scales of flow features may span several orders of magnitude. Inevitably, the smallest scales of flow cannot be captured at the resolutions used for practical fluid simulations and are lost due to both low resolution and numerical dissipation.

### 3.1 Turbulence and the Energy Cascade

Due to the chaotic and stochastic nature of fluid turbulence, a statistical approach is generally adopted both in the theory of turbulence [Monin and Yaglom 1971; McComb 1990] and in numerical methods such as Reynolds-averaged Navier-Stokes (RANS) and large eddy simulation (LES). Typically, the fluid flow is separated into a spatially or temporally averaged *mean flow*  $\mathbf{U}$  and a fluctuating component  $\mathbf{u}'$  where the turbulence resides. While the dynamics of  $\mathbf{u}'$  itself are extremely chaotic, those of its statistics such as energy are more amenable to modeling.

A popular model for the distribution of energy among different scales is Kolmogorov's famous "five-thirds law" [Frisch 1995], which states that for homogeneous, stationary turbulence, there is



**Figure 2:** Our technique runs alongside an existing fluid simulator, and has two components: an energy model for tracking turbulent energy produced and transported by the mean flow, and an advected curl noise framework for generating the turbulent component of flow.

a range of scales in which the distribution of energy  $E(k)$  over wavenumber  $k$  follows the spectrum,

$$E(k) \propto k^{-5/3} \quad (3)$$

This distribution arises due to the *energy cascade* through which turbulent energy is introduced at the *inertial scale*  $k_0$ , and cascades to smaller scales until it is dissipated by viscosity. The range of scales over which the Kolmogorov spectrum holds is called the *inertial subrange*, and the characteristics of the turbulent eddies in this range are to a large extent universal and independent of the specific large-scale flow geometry. While the reverse cascade of energy flow from small to large scales is also possible, it is in practice overwhelmingly dominated by the large-to-small cascade, so we ignore this possibility.

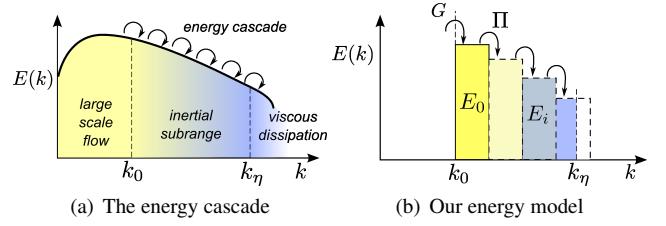
### 3.2 Overview of Our Approach

Our approach is based on the same idea of decomposing the velocity field  $\mathbf{u}$  of the fluid into a large-scale flow  $\mathbf{U}$  and a small-scale turbulent component  $\mathbf{u}'$ . We take  $\mathbf{U}$  to be the velocity computed by an existing numerical simulator. We assume that  $\mathbf{U}$  sufficiently resolves the interesting large-scale dynamics of the flow and introduce an explicit model for small-scale fluctuations  $\mathbf{u}'$ . By construction the sub-grid kinetic energy associated with  $\mathbf{u}'$  falls off with decreasing scale in a way similar to what is observed in the inertial subrange of isotropic turbulence.

We characterize the small-scale turbulent flow in terms of the distribution of its kinetic energy over space and over different scales. The two main parts of our technique, shown in Fig. 2, are (1) tracking the production, transport and dissipation of this energy over time due to the large-scale flow  $\mathbf{U}$ , and (2) synthesizing a small-scale turbulent flow field  $\mathbf{u}'$  which obeys this energy distribution. The fine-scale flow generated is used to produce highly detailed output data such as billowing smoke or fine ripples on the free surface of a liquid, and is also loosely coupled to the large-scale flow  $\mathbf{U}$ , introducing vortices and other flow detail at small grid scales.

## 4 Dynamics of Turbulent Energy

To make the problem approachable, we start with the simplifying assumptions that the turbulence is isotropic, follows the Kolmogorov spectrum at each point, and has local eddies which are uncorrelated with the large-scale flow. We divide the spectrum into octaves of spatial frequency, and define the following quantities for



**Figure 3:** The turbulent energy cascade transports energy from the mean flow through the large scales of turbulence to the smallest ones. We only track the energy  $E_0$  in the largest turbulent band, and model the rest using the Kolmogorov spectrum.

each octave  $i = 0, 1, \dots$ : length scale  $l_i = l_{i-1}/2$ , wavenumber  $k_i = l_i^{-1}$ , kinetic energy  $E_i$ , velocity magnitude  $\bar{u}_i = \sqrt{E_i}$ , and eddy turnover time  $\tau_i = l_i/\sqrt{u_i}$ . Now the kinetic energy  $E_0$  in the lowest octave of turbulent eddies fully defines the energy in the  $i$ th octave simply as

$$E_i = E_0(k_i/k_0)^{-2/3} \quad (4)$$

The change in exponent from  $-5/3$  to  $-2/3$  from Eq. 3 is because we are now considering the energy in one *octave*, which we obtain by integrating the energy per unit wavenumber over an octave range. We allow  $E_0$ , and thus  $E_i$ , to be spatially varying scalars in order to represent the distribution of turbulence in complex inhomogeneous flow. The magnitude of the eddy velocities at any scale are then on the order of  $\bar{u}_i$ , which we will directly use to generate a spatially modulated velocity field.

There is no closed solution for describing the behavior of the turbulent kinetic energy; nevertheless, on physical grounds, the important physical phenomena involved in the energy dynamics of turbulence are as follows: (1) production of turbulence via forcing from the mean flow, (2) advection by the mean flow, (3) diffusion of kinetic energy due to turbulent mixing, (4) cascade of energy from larger to smaller scales due to non-linear self-advection, and (5) viscous dissipation, which removes energy at the smallest scale of eddies. Thus, in abstract terms, we can express the dynamics of energy at the inertial scale as

$$\frac{\partial E_0}{\partial t} + \mathbf{U} \cdot \nabla E_0 = G(E_0, \mathbf{U}) - \Pi(E_0) + D(E_0) \quad (5)$$

where  $G$  is the production term through which turbulent energy is generated,  $\Pi$  is the rate of energy flow through the cascade, and  $D$  describes the spatial diffusion of energy due to turbulent mixing. We ignore dissipation since viscosity is only significant at the smallest scales. Fig. 3 shows how this approach relates to the theoretical energy cascade.

If these terms are estimated reasonably, the behavior of the turbulent eddies will look plausible. We now describe our approximations for the magnitudes of these quantities to achieve this.

### 4.1 Energy Transfer

First, the inertial scale  $k_0$  must be chosen to define the size of the largest turbulent eddies modeled. In our experiments, we chose the eddy size  $l_0 = k_0^{-1}$  to be a small multiple of the grid spacing of the numerical simulator. Overlapping the scales modeled numerically and procedurally in this way has two advantages: it allows the turbulent flow to compensate for flow details lost to numerical diffusion, and it makes it possible for the turbulence to affect the background flow at the overlapping scales.

**The production term** Through  $G$ , the mean flow acts as a driving force which creates and intensifies turbulent energy. The simplest approximation that can be made here is that turbulence acts as a viscous drag on the mean flow, characterized by an *eddy viscosity*  $\nu_t$  and taking energy at the rate  $\nu_t S(\mathbf{U})^2$ , where  $S(\mathbf{U})^2$  is the squared norm of the strain rate. The eddy viscosity can further be approximated through mixing-length theory as  $\nu_t = l_0 u_0 = k_0^{-1} \sqrt{E_0}$  [Davidson 2004]. This gives the net rate of energy production as

$$G(E_0, \mathbf{U}) = k_0^{-1} \sqrt{E_0} S(\mathbf{U})^2 \quad (6)$$

where formally  $S(\mathbf{U})^2 = \sum \sum ((\partial U_i / \partial x_j + \partial U_j / \partial x_i) / 2)^2$ .

**The cascade term** A simple estimate can be obtained using the conjecture (from Obukhov, via [Davidson 2004]) that turbulent eddies at the inertial scale pass on energy into smaller ones on the timescale  $\tau_0$ . Since the energy carried by these eddies is  $E_0$ , the rate of energy transfer becomes

$$\Pi(E_0) = E_0 / \tau_0 = k_0 E_0^{3/2} \quad (7)$$

**The diffusion term** Kinetic energy at a single octave diffuses over space as turbulent eddies transport fluid, causing momentum and energy transfer. By analogy with the turbulent mixing of a passive scalar being advected by the turbulent flow, whose diffusion rate is roughly  $\alpha \propto l_0 \bar{u}_0$  [Davidson 2004], we describe the turbulent diffusion of energy as

$$D(E_0) = \nabla \cdot (\alpha \nabla E_0) = \nabla \cdot (l_0 \sqrt{E_0} \nabla E_0) \quad (8)$$

**Viscosity** The role of viscosity in turbulent flow is only significant at the smallest eddy scales, and is limited to dissipating the cascading energy at the highest wavenumbers. To reproduce this, we simply cut off the energy spectrum at the scale where the rate of viscous dissipation equals the cascade rate  $\Pi$ . This *Kolmogorov microscale*  $k_\nu$  is given by [McComb 1990]

$$k_\nu = \left( \frac{\Pi(E_0)}{\nu^3} \right)^{1/4} \quad (9)$$

After the energy  $E_0$  at the largest octave is known, we can define the energy  $E_i$  at any other octave  $i$  with reciprocal length scale  $k_i$  using a Kolmogorov spectrum cut off at  $k_\nu$ , as Fig. 3 illustrates. If the cutoff falls within an octave, we scale that octave's energy by how much of it lies before the cutoff. This approach allows the model to “turn off” turbulence appropriately when the flow is calm, as  $k_\nu$  will become lower than the first octave  $k_0$  itself, and no energy will be present in the subgrid scales.

$$E_i = \begin{cases} E_0 \left( \frac{k_i}{k_0} \right)^{-2/3} & \text{if } 2k_i < k_\nu \\ E_0 \left( \frac{k_i}{k_0} \right)^{-2/3} \frac{2k_i - k_\nu}{2k_i - k_i} & \text{if } k_i \leq k_\nu \leq 2k_i \\ 0 & \text{if } k_\nu < k_i \end{cases} \quad (10)$$

## 4.2 Numerical Issues

The dynamics of turbulent kinetic energy as given in Eq. 5 essentially form an advection-reaction-diffusion PDE. As in [Kim and Lin 2007], we split the integration into diffusion, advection and reaction stages, allowing us to maintain stability at each step.

Since the turbulent energy  $E_0$  is governed by the large-scale flow  $\mathbf{U}$  from the numerical method, we represent it on the same coarse grid. This allows us to take the same large timesteps as the numerical

simulation. The advection step can be performed using the same scheme as used in the fluid solver, in our case using BFECC.

For the diffusion step, the nonlinearity in the diffusion rate presents numerical difficulties at regions where  $E_0$  has large changes in magnitude. In the interest of stability and efficiency, we replace it with a constant spatially varying term which is filtered to avoid the degeneracy, as follows:

$$\tilde{D}(E_0) = k_0^{-1} \nabla \cdot \left( \sqrt{\tilde{E}_0^t} \nabla E_0^{t+1} \right) \quad (11)$$

where  $\tilde{E}_0^t$  is a filtered version of  $E_0^t$  obtained by Gaussian filtering with a radius  $l_0$ . This spatially varying but linear diffusion equation can then be integrated efficiently by an alternate dimensions implicit (ADI) method [Kass et al. 2006]. This provides a fast approximate solution which is unconditionally stable, and acceptably accurate as long as  $\bar{u}_0 < l_0 / \Delta t$ .

The turbulence production and cascade terms require no neighbor information, and so can be solved as a simple pointwise ODE, with  $E'_0 = G(E_0, \mathbf{U}) - \Pi(E_0) = f(E_0)$ . This ODE has a unique stable equilibrium for positive  $E_0$ . Large timesteps can be taken with stability by choosing between forward Euler or backward Euler timestepping based on the sign of  $df/dE_0$ .

The initial value of  $E_0$  should not be precisely zero since the production term  $G$  contains a factor with  $E_0$  itself. However, in practice, a very small value such as  $10^{-4}$  can be chosen; this does not affect the characteristics of the results, which are dominated by the interaction of the production and cascade terms.

**A note on boundary layers** Physically, a solid boundary immersed in fluid imposes a no-slip condition on the flow, producing a thin boundary layer over which the relative velocity of the fluid ramps down to zero. It is here that turbulence due to obstacles is generated. Explicitly accounting for such boundary layers would add significantly to the implementation complexity of our model, and the extremely high strain rate and extremely small thickness of the layer could lead to numerical difficulties. Instead, in the current work, we have simply accounted for the no-slip condition by using a ghost velocity past the boundary equal that of the obstacle when computing  $\mathbf{S}(\mathbf{U})$ .

## 5 Procedural Synthesis of Turbulent Flow

Having defined the distribution of turbulent energy over space and on different scales, it remains to actually generate an incompressible, turbulent velocity field  $\mathbf{u}'$  which has the prescribed energy distribution. The eddies and vortices in this velocity field must also move with the large-scale flow, and fluctuate over time for a realistic appearance of turbulence. In this section, we describe our method to generate such a velocity field which has these properties.

Curl noise [Kniss and Hart 2004; Bridson et al. 2007] uses the vector calculus identity  $\nabla \cdot (\nabla \times \psi) = 0$  to provide an efficient means of generating incompressible flow fields supporting spatial modulation and rigid boundaries. Several octaves of such curl noise can be added, each modulated by the desired velocity magnitude  $\bar{u}_i$ , to generate an instantaneous velocity field with the desired energy spectrum. However, the temporal behavior of the velocity field over time is also important. First, the turbulent velocity field should be advected by the mean flow  $\mathbf{U}$ , in order for the turbulent eddies to be seen as being carried along by the fluid. Second, the turbulent velocities should also fluctuate over time at a rate based on the eddies' characteristic timescale, given by  $\tau_i$  which varies over space.

Neyret [2003] has presented a method for advecting and regenerating textures to avoid distortion; this has been used in concurrent work [Kim et al. 2008b] for advecting wavelet turbulence. In our experiments, we had found that textures had to be regenerated too often in regions of high strain rate, and the use of texture coordinates made looking up  $\tau_i$  difficult. We take a simpler Lagrangian approach, representing the potential function using basis functions centered on particles which are advected with the flow. Such noise particles had been used previously by Stam and Fiume [1993] for advecting smoke, and Zhu and Bridson [2005] for gritty sand rendering. In combination with curl noise, such an approach allows for efficient, distortion-free transport of incompressible vector noise.

### 5.1 Noise Particles

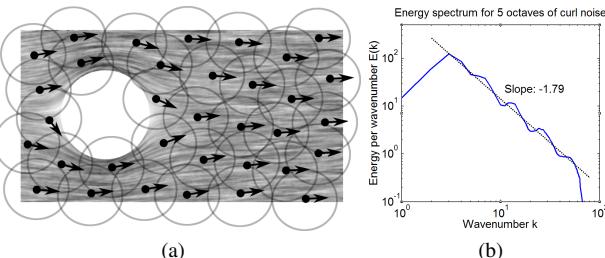
To be precise, we represent a vector potential field  $\psi$  as the sum of volumetric noise textures centered at a set of particles  $\mathbf{p}_j$  distributed uniformly within the fluid, as illustrated in Fig. 4(a). A smooth radial falloff  $\rho(r)$  which goes to zero at  $r = l_0$  is used for blending between particles. For noise at a particular scale  $i$ , each particle carries an animated vector noise function  $\mathbf{N}_j^i(\mathbf{r}, t_i)$  of the appropriate scale. The net potential for one octave,  $\psi_i$ , at a point  $\mathbf{x}$ , modulated by a spatially varying amplitude  $A_i$ , is then given by the weighted average of scaled noise functions,

$$\psi_i(\mathbf{x}) = \frac{\sum_j A_i(\mathbf{p}_j) \rho(||\mathbf{x} - \mathbf{p}_j||) \mathbf{N}_j^i(\mathbf{x} - \mathbf{p}_j, t_i)}{\sum_j \rho(||\mathbf{x} - \mathbf{p}_j||)} \quad (12)$$

We take moving boundaries into account by further modulating  $\psi_i$  using the approach of Bridson et al. [Bridson et al. 2007]. The derived velocity field is simply  $\mathbf{u} = \nabla \times \psi = \nabla \times \sum_i \psi_i(\mathbf{x})$ . Some of the results we show later are from our initial implementation, in which we had used a weighted sum neglecting the normalization in the denominator of  $\psi_i$ , but it was pointed out by a reviewer that this could potentially cause spurious fluctuations.

For the radial falloff  $\rho(r)$ , we require a kernel which goes smoothly to zero at  $r = l_0$  and has continuous derivatives; the cubic spline  $1 - 3r^2 + 2r^3$  suffices. For different octaves, the amplitude is chosen as  $A_i = \bar{u}_i l_i$  so that the derived velocity field has magnitude  $\bar{u}_i$ . The 3 components of the vector noise functions  $\mathbf{N}^i$  were taken as different offsets of animated 3D Perlin noise, which we precomputed. Wavelet noise [Cook and DeRose 2005] could be used for better band-limiting and isotropy characteristics, as suggested by Kim et al. [2008b]. We found that the use of Perlin noise nevertheless has reasonable spectral behavior as shown in Fig. 4(b), and does not really harm the visual plausibility of the result.

Initially, the particles are distributed over the volume of the fluid in a Poisson-disk distribution, with a distance of  $l_0$  to allow the RBFs  $\rho$  to overlap. Advection from the mean flow  $\mathbf{U}$  is taken into account by simply transporting the particles forward with the flow. This



**Figure 4:** (a) Noise particles are distributed uniformly in the fluid volume and advected by the large-scale flow. (b) The energy spectrum of turbulent flow generated by 5 octaves of curl noise shows a power law with exponent close to the expected value of  $-1.66$ .

has the effect of transporting the derived velocity field  $\mathbf{u}$  along the flow as well, while also maintaining incompressibility and avoiding anisotropic distortion which would not have been the case had we advected  $\mathbf{u}$  directly. A dynamic sampling scheme could also be used to delete and insert particles to maintain an approximate Poisson-disk distribution in 3D [Bridson 2007], but we did not find this necessary.

It remains to make the generated velocity field exhibit temporal fluctuations on the timescale of the eddy turnover time  $\tau_i$ . Recall that each particle carries an index into an animated noise function, whose timescale we can choose such that it fluctuates, say, once every unit  $t$ . Then we advance the time coordinate  $t_i$  of the noise function by  $\tau_i^{-1}$  per second, which has the desired effect.

### 5.2 Coupling to the Large-scale Flow

Finally, we describe the method by which the generated turbulent flow affects the large-scale fluid flow handled by the fluid simulator. Unlike [Selle et al. 2005], since we keep the turbulent flow represented by particles and the large-scale simulated flow separate, we do not use the vorticity confinement force to drive the simulated fluid. Instead, we simply use the turbulent velocity field  $\mathbf{u}'$  as an additional velocity when performing the self-advection step in the simulation. That is, we advect the simulation velocity  $\mathbf{U}$  not just by itself but by the sum  $\mathbf{U} + \mathbf{u}'$ , appropriately resampled to the simulation grid. This simple coupling models the effects of turbulent advection on the coarse grid which is visually important.

Similarly, for the advection of scalar quantities, such as density for smoke animations, we use the same net velocity field  $\mathbf{U} + \mathbf{u}'$ . The scalar quantities and the final turbulent velocity are represented on finer grids than the simulation, allowing high-resolution detail to be obtained without increasing the computational cost of the numerical simulation. The density has to be downsampled at each step to compute the forcing effects such as buoyancy forces on the simulation, but this is a cheap operation. Since it is such scalar quantities that are directly visualized and not the fluid velocity field itself, this approach provides an immediate visual benefit.

### 5.3 Free Surfaces

A similar approach can be applied to the animation of liquids, where by representing the free surface at high resolution, fine surface details such as small ripples can be obtained. However, a fundamental issue is that in the presence of a free surface, the isotropy and homogeneity assumptions of the Kolmogorov model break down, and the turbulent surface features show a complex dynamical behavior for which “the present state of the art [in fluid dynamics] is quite controversial” [Magnaudec 2003]. In the absence of a single well-accepted theory of free-surface turbulence, we have chosen to extend our Kolmogorov-based model as a reasonable compromise to maintain the coarse simulation grid and synthesize the fine-scale behavior procedurally, although it falls well outside the model assumptions. We present this as initial work on the topic of free-surface turbulence for computer animation, in the hope that it will encourage further exploration.

First of all, since the free surface also defines the simulation domain for the numerical method, a conservative downsampling is necessary to ensure known values of fluid velocity throughout the fluid. We mark a simulation cell as fluid if any fluid is present in it in the high resolution surface. To take into account the free surface fluctuations, it can be modeled as a variable density flow, with the density equated to the filled volume fraction. For this, the pressure projection equation is modified to be  $\mathbf{U} = \mathbf{U}^* - (\nabla p)/\rho$ , and therefore

$\nabla \cdot \mathbf{U}^* = \nabla \cdot (\nabla p / \rho)$  as described in [Kim et al. 2007], where  $\mathbf{U}^*$  is the pre-projection velocity field.

Another problem is that the simulation is unaffected by the high-resolution ripples introduced on the fluid surface. This means that subgrid-scale surface ripples have no dynamical effect and do not die out over time, making the fluid appear to be viscoelastic. We counteract this by introducing two additional effects to make the high-resolution ripples, if present, continue to move and fluctuate, and gradually die out over time. They do not exactly correspond to the equations of wave dynamics, however; attempting to simulate the wave nature of subgrid-scale ripples would tie the timestep to the very small grid size of the high-res surface, which is something we have avoided throughout the paper.

If the surface is represented as the level set of a function  $\phi$ , we compute a smoothed version  $\tilde{\phi}$  in an efficient manner by filtering and downampling it to the coarse simulation grid, and reinitializing it by a standard fast marching method, e.g. [Losasso et al. 2005].  $\tilde{\phi}$  represents a smooth surface with subgrid-scale features removed. We model the decay of the small-scale features by timestepping  $\phi$  to approach  $\tilde{\phi}$  over a timescale  $\tau'$ .

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{\tau'} (\tilde{\phi}^n - \phi^n) \quad (13)$$

We define the characteristic timescale  $\tau'$  of subgrid features in terms of the simulation grid resolution  $\Delta x$  as  $\tau' = \Delta x / c$ , where  $c = \sqrt{g\Delta x / (2\pi)}$  is the speed of deep water gravity waves of wavelength  $\Delta x$ . Surface tension only becomes significant for waves of wavelength smaller than 1.7 cm [Lamb 1993]; we have neglected this in the current work.

Secondly, the characteristic “rippling” motion can be obtained by ensuring that there is always some turbulent energy associated with the high-resolution features which will cause them to move and fluctuate. We define the surface turbulent energy associated with the ripples as

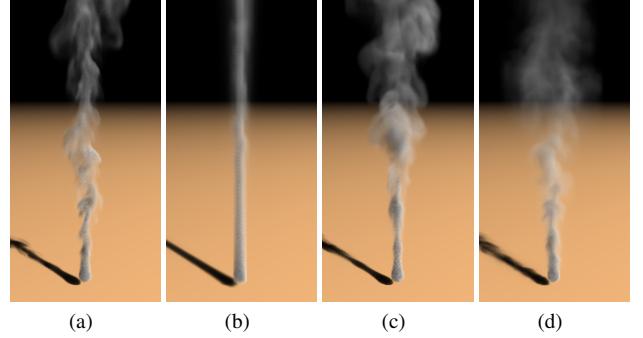
$$E' = \bar{u}^2 = (\delta / \tau')^2 \quad (14)$$

where  $\delta$  is the average height of the ripples.  $\delta$  can be estimated as the average magnitude of  $\phi - \tilde{\phi}$  in the grid cell. Using  $\max(E_0, E')$  instead of  $E_0$  as the turbulent energy distribution at each point, we achieve plausible motion of ripples in a simple and efficient manner.

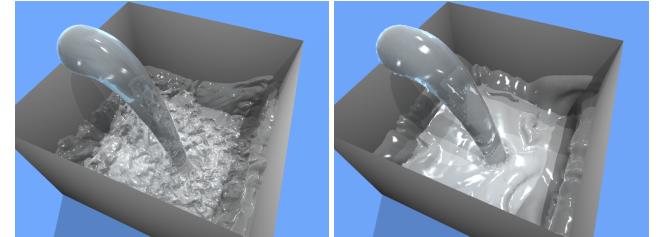
## 6 Results

In our implementation, we used a particle level set (PLS) solver [Bridson and Müller-Fischer 2007] with BFECC [Kim et al. 2007] for the numerical simulation of the fluid. In all scenarios, we used a  $4 \times 4 \times 4$  times finer resolution for the turbulence as compared to the simulation grid resolution for the fluid solver. We found this  $4 \times$  refinement to be adequate to resolve visually relevant details in our scenarios; higher values would produce finer detail (as long as the intensity of turbulence is high enough) but would not otherwise change the characteristics of the result.

The primary control parameter which defines the behavior of the fluid is its viscosity  $\nu$ . This controls the viscous cutoff  $k_\nu$  in Eq. 10. Higher values would delay the onset of turbulence and prevent very fine eddies from forming, as appropriate for a more viscous fluid. The other parameter that can be varied is the size of the largest eddies,  $l_0$ . This is a more artistic parameter that controls the scale of the injected turbulent flow. Letting the simulation grid size be  $h$ , we used  $l_0 = 4h$  in all our examples, except the rocket launch simulation which used  $2h$  to obtain relatively smaller eddies.



**Figure 5:** Smoke rising from a small density source, simulated with (a) our method, (b) low-resolution simulation only, (c) reference simulation on a  $4 \times$  refined grid, and (d) vorticity confinement on  $2 \times$  refinement.



**Figure 6:** Water pouring into a tank creates a chaotic, turbulent surface. These fine surface details are reproduced by our method (left), while the base simulation can only generate a very smooth fluid surface (right).

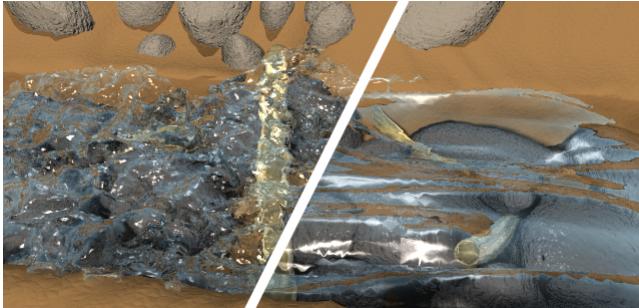
### 6.1 Benchmarks

Fig. 5 shows smoke rising from a spherical density source due to buoyancy forces. The smoke column is only a few grid cells across, and BFECC is not able to recover any detailed vorticity. Our method produces visually plausible results and reproduces the transition from laminar to turbulent flow as flow velocity increases, thanks to the viscous cutoff at  $k_\eta$ . This would be difficult to achieve with a vortex particle method. We also show for comparison the results of vorticity confinement [Fedkiw et al. 2001]. On the original low-resolution grid, it had little effect; we show a simulation on a  $2 \times$  refined grid, which exhibits turbulent behavior but is only biased towards the finest scale eddies.

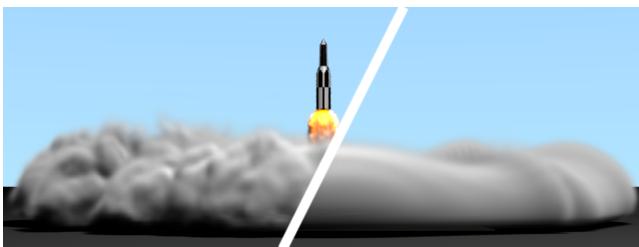
We demonstrate our free surface handling approach in Fig. 1(b). Our method captures very high-resolution features in the wake of the moving ball, which appropriately die down over time as the driving mean flow is removed. More free-surface flows are shown in the water pouring into box (Fig. 6) and rivers rapids flowing over the bed rocks and tree logs (Fig. 7(a)). In all cases, our results are rendered with the high-resolution surface while the base simulation has only the coarse level set.

In Fig. 7(b), we simulate the turbulent exhaust jet during a rocket launch. The turbulent eddies show plausible motion in both the jet itself and the billowing smoke on the ground. The rocket launch simulation used 98,000 noise particles due to the choice of relatively smaller  $l_0$ . All other simulations used between 6,100 and 12,300 noise particles.

In the supplemental video, we show comparisons with the basic BFECC simulation without adding turbulence. For smoke, a fairer comparison is done by passively advecting high-resolution density under the coarse-resolution flow field; this reduces diffusion of den-



(a) River rapids



(b) Rocket launch

**Figure 7:** (a) River rapids show turbulence due to fast-moving flow in a complex channel with many obstacles. (b) Simulation of turbulent exhaust jet during a rocket launch. In both figures, our results are on the left, while the results of fluid simulation without added turbulence are on the right.

sity, but it cannot improve the dynamics of the flow itself.

## 6.2 Performance and Comparison

Our method is an order of magnitude faster than a full numerical simulation at 4x higher resolution. The time for tracking the energy distribution was negligible. The bottleneck in our approach turned out to be the evaluation of the potential in advected curl noise, because this step involves irregular memory accesses on a very high-resolution grid. Timings were taken on an Intel Xeon 2.8 GHz processor with 8 GB of RAM. Fig. 8 summarizes the performance of our method in comparison with basic low- and high-resolution simulations. We did not perform liquid simulations with high-resolution advection alone because of the free-surface issues discussed in 5.3.

Scene	Grid size	Ours	High-res	Gain
Smoke (Fig.1a)	$256 \times 32 \times 32$	56 sec	1192 sec	21x
Wakes (Fig.1b)	$128 \times 32 \times 32$	45 sec	708 sec	16x
Box (Fig. 6)	$64 \times 64 \times 64$	70 sec	360 sec	5x
Rapids (Fig. 7(a))	$150 \times 30 \times 50$	105 sec	776 sec	7x
Launch (Fig. 7(b))	$64 \times 64 \times 64$	142 sec	3186 sec	24x

**Figure 8:** Performance comparison of our method with a full high-res simulation needed to generate the same visual detail without added turbulence. All timings are per-frame and include the time taken by the fluid solver, the turbulence model, and scalar advection.

## 6.3 Analysis and Discussion

The results demonstrate the flexibility and generality of our method. It replaces the computational complexity of high-resolution fluid

simulation with a simple turbulence model, giving a major performance benefit since fluid simulation on a coarser grid can take much larger timesteps than it could at the full fine-scale resolution. The turbulence model itself is stable and can take equally large timesteps as the coarse fluid simulation allows. There is only one global solve, which appears in the diffusion step and is performed efficiently through dimensional splitting.

We found the main bottlenecks in the speed of our approach to be the evaluation of the noise function, and the maintenance of the free surface. A comparison of our advected noise with Neyret’s algorithm [Neyret 2003] in terms of quality and performance is left for future work. For the free surface, the level set needs to be reinitialized at each step, which causes a performance hit.

Interesting questions remain about how much the scales of procedural turbulence and the simulated flow should overlap. Less dissipative numerical simulations as proposed in recent work would likely not need as much fluctuations to be injected at the grid level, so  $l_0$  might be chosen to be as small as the grid size. Further, whether the combined spectrum of the large-scale and turbulent flow has an appropriate shape may be worth investigating. We leave these experiments for further investigation.

Finally, the applicability of the Kolmogorov model depends on the assumptions of isotropy and homogeneity, and as such, the model is being extended beyond its strict bounds to model the complex, directed flows seen in animation. Our surface handling approach is relatively simplistic and cannot guarantee volume conservation. However, the complexity of the phenomenon of turbulence is such that physical accuracy is unachievable without great computational expense. We have deliberately chosen such an approach to approximate the behavior of turbulent fluids, so that a simple and efficient technique is possible for generating plausible visual effects. Much research is still needed in modeling anisotropy, higher-order statistics, detailed free-surface phenomena, and other properties of true physical turbulence without an exorbitant computational cost.

## 7 Conclusion

We have presented a fast and effective technique for simulating turbulent fluids through a subgrid turbulence model that can be integrated into existing simulations and tracks the production and evolution of turbulent flow automatically. We show that this approach is capable of preserving the small-scale eddies and vortices often lost due to numerical dissipation and maintains a loose coupling between the large-scale flows and subgrid-resolution turbulence. We are able to achieve the sub-grid resolution details equivalent to those of a significantly finer-resolution fluid simulation at only a small fraction (roughly one order of magnitude less) of computational cost. This approach is general and applicable to a wide variety of fluid phenomena as demonstrated, including swirling smoke, roiling exhaust, rippling water wakes, and rapid turbulent flows.

**Acknowledgements** We would like to thank Ted Kim, Ron Henderson and the anonymous reviewers for their valuable suggestions. This research was supported in part by the Army Research Office, the National Science Foundation, Intel, and RDECOM.

## References

- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. *ACM Trans. Graph* 26, 3, 48.
- ANGELIDIS, A., AND NEYRET, F. 2005. Simulation of smoke based on vortex filament primitives. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics As-

- sociation, Los Angeles, California, D. Terzopoulos and V. Zordan, Eds., 87–96.
- BRACKBILL, J. U., AND RUPPEL, H. M. 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J. Comput. Phys.* 65, 2, 314–343.
- BRIDSON, R., AND MÜLLER-FISCHER, M. 2007. Fluid simulation: SIGGRAPH 2007 course notes. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, ACM, New York, NY, USA, 1–81.
- BRIDSON, R., HOURIHAN, J., AND NORDENSTAM, M. 2007. Curl-noise for procedural fluid flow. *ACM Trans. Graph* 26, 3, 46.
- BRIDSON, R. 2007. Fast Poisson disk sampling in arbitrary dimensions. In *SIGGRAPH '07: ACM SIGGRAPH 2007 sketches*, ACM, New York, NY, USA, 22.
- CHENTANEZ, N., FELDMAN, B. E., LABELLE, F., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *Symposium on Computer Animation*, Eurographics Association, M. Gleicher and D. Thalmann, Eds., 219–228.
- COOK, R. L., AND DEROSSE, T. 2005. Wavelet noise. *ACM Trans. Graph.* 24, 3, 803–811.
- DAVIDSON, P. A. 2004. *Turbulence: An Introduction for Scientists and Engineers*. Oxford University Press.
- ELCOTT, S., TONG, Y., KANSO, E., SCHRÖDER, P., AND DESBRUN, M. 2007. Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph* 26, 1.
- FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *SIGGRAPH*, 15–22.
- FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *SIGGRAPH*, 23–30.
- FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical Models and Image Processing* 58, 5, 471–483.
- FRISCH, U. 1995. *Turbulence: The Legacy of A. N. Kolmogorov*. Cambridge University Press.
- KASS, M., LEFOHN, A., AND OWENS, J. 2006. Interactive depth of field. Tech. Rep. 06-01, Pixar Animation Studios.
- KIM, T., AND LIN, M. C. 2007. Stable advection-reaction-diffusion with arbitrary anisotropy. *Journal of Visualization and Computer Animation* 18, 4-5, 329–338.
- KIM, B., LIU, Y., LLAMAS, I., AND ROSSIGNAC, J. 2007. Advections with significantly reduced dissipation and diffusion. *IEEE Trans. Vis. Comput. Graph* 13, 1, 135–144.
- KIM, D., SONG, O.-Y., AND KO, H.-S. 2008. A semi-Lagrangian CIP fluid solver without dimensional splitting. *Computer Graphics Forum* 27, 467–475(9).
- KIM, T., THÜREY, N., JAMES, D., AND GROSS, M. 2008. Wavelet turbulence for fluid simulation. In *SIGGRAPH*.
- KNISSL, J., AND HART, D., 2004. Volume effects: modeling smoke, fire, and clouds. Section from ACM SIGGRAPH 2004 courses, *Real-Time Volume Graphics*, [http://www.cs.unm.edu/~jmk/sig04\\_modeling.ppt](http://www.cs.unm.edu/~jmk/sig04_modeling.ppt).
- LAMB, H. 1993. *Hydrodynamics*. University Press.
- LOSASSO, F., FEDKIW, R., AND OSHER, S. 2005. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids* 35, 2006.
- MAGNAUDET, J. 2003. High-Reynolds-number turbulence in a shear-free boundary layer: revisiting the HuntGraham theory. *Journal of Fluid Mechanics* 484, 167–196.
- MCCOMB, W. D. 1990. *The Physics of Fluid Turbulence*. Oxford University Press.
- MONIN, A. S., AND YAGLOM, A. M. 1971. *Statistical Fluid Mechanics*. Dover Publications.
- MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, Eurographics Association, San Diego, California, D. Breen and M. Lin, Eds., 154–159.
- NEYRET, F. 2003. Advectored textures. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, Eurographics Association, San Diego, California, D. Breen and M. Lin, Eds., 147–153.
- PARK, S. I., AND KIM, M. J. 2005. Vortex fluid for gaseous phenomena. In *ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, Eurographics Association, Los Angeles, California, D. Terzopoulos and V. Zordan, Eds., 261–270.
- PERLIN, K., AND NEYRET, F. 2001. Flow noise. In *Siggraph Technical Sketches and Applications*, 187.
- PREMOZE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A. E., AND WHITAKER, R. T. 2003. Particle-based simulation of fluids. *Comput. Graph. Forum* 22, 3, 401–410.
- SCHECHTER, H., AND BRIDSON, R. 2008. Evolving sub-grid turbulence for smoke animation. In *Eurographics/SIGGRAPH Symposium on Computer Animation*.
- SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. *ACM Transactions on Graphics* 24, 3 (July), 910–914.
- SELLE, A., FEDKIW, R., KIM, B., LIU, Y., AND ROSSIGNAC, J. 2008. An unconditionally stable MacCormack method. *Journal of Scientific Computing*. (in press).
- SHINYA, M., AND FOURNIER, A. 1992. Stochastic motion - motion under the influence of wind. *Computer Graphics Forum* 11, 3 (Sept.), 119–128. EG92: Cambridge, UK., Editors: A. Kilgour and L. Kjelldahl.
- SIMS, K. 1990. Particle animation and rendering using data parallel computation. In *Computer Graphics, Proceedings of Siggraph*, ACM, vol. 24, 405–413.
- STAM, J., AND FIUME, E. 1993. Turbulent wind fields for gaseous phenomena. In *SIGGRAPH*, ACM, 369–376.
- STAM, J. 1999. Stable fluids. In *SIGGRAPH*, 121–128.
- WEJCZERT, J., AND HAUMANN, D. 1991. Animation aerodynamics. T. W. Sederberg, Ed., vol. 25, 19–22.
- ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, 965–972.