Curl-Noise for Procedural Fluid Flow

Robert Bridson* University of British Columbia
Jim Hourihan† Tweak Films
Marcus Nordenstam‡ Double Negative

Abstract

Procedural methods for animating turbulent fluid are often preferred over simulation, both for speed and for the degree of animator control. We offer an extremely simple approach to efficiently generating turbulent velocity fields based on Perlin noise, with a formula that is exactly incompressible (necessary for the characteristic look of everyday fluids), exactly respects solid boundaries (not allowing fluid to flow through arbitrarily-specified surfaces), and whose amplitude can be modulated in space as desired. In addition, we demonstrate how to combine this with procedural primitives for flow around moving rigid objects, vortices, etc.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: noise, turbulence, fluids, procedural animation

1 Introduction

Many shots in films and effects in games call for fluid-like turbulent motion, particularly for smoke and vapor. Though simulation of the equations of fluid motion can generate spectacular animation, it can be frustratingly slow and unwieldy to direct. Many practitioners instead turn to procedural methods where the state of the system, such as the velocity field of the fluid, can be cheaply and repeatably evaluated anywhere in space and time—without discretizing PDE's, without large grids, without simulation parameters to tweak, without solving systems of equations, and with immediate and direct animator control. After reviewing some previous procedural methods and their drawbacks, we offer a new, fast and simple procedural approach for constructing fluid-like velocity fields.

Both Sims [1990] and Wejchert and Haumann [1991] used a linear superposition of "flow primitives" such as vortices, sources, sinks, and particular solutions of potential flow to generate plausible wind velocity fields. However, without manually adding many vortices, this approach is restricted to fairly laminar flow, and matching the flow to arbitrary solid geometry is not handled.

Shinya and Fournier [1992] and Stam and Fiume [1993] used Fourier synthesis to produce physically plausible turbulent velocity fields. However, arbitrary solid boundaries cannot be handled with this method, the entire 3D domain must be computed and stored (particularly difficult in the constrained memory environment of game consoles), and artist control such as spatially modulating the magnitude of the turbulence is lost. Later, Stam [1997] showed that if just a few point samples of wind velocity are required (as in his modal tree dynamics) there is no need for storing and calculating the full 3D domain; however, the approach is inefficient for large numbers of samples (e.g. when advecting large numbers of particles), and doesn't solve the problem of modulating Fourier synthesized fields.

*e-mail: rbridson@cs.ubc.ca
†e-mail: jimh@tweakfilms.com
‡e-mail: mkn@dneg.com

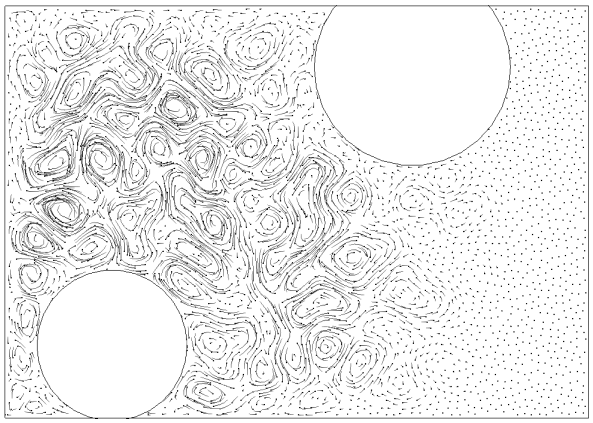

Figure 1: *Incompressible 2D noise with solid boundaries. To compute the potential ψ we multiply scaled noise $N(\vec{x},t)$ by a modulation function (a smoothed step function A of distance from the mouse cursor) and a ramp to zero based on distance $d(\vec{x})$ to the closest solid boundary: $\psi(\vec{x},t) = \text{ramp}\left(\frac{d(\vec{x})}{d_0}\right) A(\vec{x}) N\left(\frac{\vec{x}}{d_0},t\right)$. The velocity field is the curl of this potential: $\vec{v} = \nabla \times \psi$.*

Perlin's eponymous noise function [1985; 2002] is frequently used in practice to generate random velocity fields; however, these fields generally contain many sinks ("gutters" where particles accumulate) since they are not divergence-free. The divergence-free condition, $\nabla \cdot \vec{v} = 0$, is equivalent to stating the fluid is incompressible, and is one extremely important visual characteristic of everyday fluids (e.g. water, air, and smoke).

Perlin and Neyret [2001] also created time-varying textures which appear to flow, but cannot be used for moving particle systems and cannot naturally handle arbitrary solid geometry.

Lamorlette and Foster [2002] use procedural methods including a Fourier-synthesized turbulence model to animate flames, and also provide good argument as to why procedural methods can be preferable to fluid simulation.

Kniss and Hart [2004] demonstrated the idea of using the curl of Perlin noise (as we do) for incompressible flow fields; our paper extends this to handle boundary conditions and other effects.

Patel and Taylor [2005] introduce "fast simulation noise", a divergence-free velocity field that can be evaluated similar to Perlin noise. While similar in spirit to our method, it suffers from either a lack of smoothness or periodic dead spots (points of zero velocity) and it cannot yet handle arbitrary solid boundaries as we do below.

Finally von Funck et al. [2006] build divergence-free velocity fields for shape deformation with a slightly different construction. While it is not clear how to adapt this approach to handle boundaries, it could in principle also be used for our application.

2 The Method

2.1 Curl

In a nutshell, we use the curl $\nabla\times$ of a potential field ψ for velocities. In three dimensions, the potential is a vector-valued field $\vec{\psi} = (\psi_1, \psi_2, \psi_3)$, giving:

$$\vec{v}(x,y,z) = \left(\frac{\partial \psi_3}{\partial y} - \frac{\partial \psi_2}{\partial z}, \frac{\partial \psi_1}{\partial z} - \frac{\partial \psi_3}{\partial x}, \frac{\partial \psi_2}{\partial x} - \frac{\partial \psi_1}{\partial y} \right) \quad (1)$$

and in two dimensions the potential is a simple scalar field, giving:

$$\vec{v}(x,y) = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}, \right). \quad (2)$$

We recall from fluid dynamics that the potential in 2D may be called the "stream function": its isocontours are the streamlines of the flow.

A classic vector calculus identity is that the curl of a smooth potential is automatically divergence-free: $\nabla \cdot \nabla\times \equiv 0$. Thus the velocity field we have constructed is divergence-free, $\nabla \cdot \vec{v} = 0$, i.e. it is incompressible. No sources or sinks ("gutters") are possible.

To evaluate the partial derivatives, we use simple finite difference approximations with a very small displacement (e.g. in our examples we have used a displacement 10^{-4} times smaller than the domain, which works fine in single precision); this makes it easy to use even quite complicated potentials.

2.2 Noise

To construct a randomly varying velocity field we use Perlin noise $N(\vec{x})$ in our potential. In 2D, $\psi = N$. In 3D we need three components for the potential: three apparently uncorrelated noise functions (a vector $\vec{N}(\vec{x})$) do the job, which in practice can be the same noise function evaluated at large offsets.

Note that if the noise function is based on the integer lattice and smoothly varies in the range $[-1,1]$, then the partial derivatives of the scaled $N(x/L)$ will vary over a length-scale L with values approximately in the range $O([-1/L, 1/L])$. This means we can expect vortices of diameter approximately L and speeds up to approximately $O(1/L)$: the user may use this to scale the magnitude of ψ to get a desired speed.

The usual trick of adding several octaves at different scales together to get "turbulent noise" (in the graphics sense [Perlin 1985]; see also Cook and DeRose's wavelet noise [2005]) actually, in this case, produces something quite similar to *physical* turbulence. Using a power law to reduce the magnitude of velocities from smaller-scale vortices, as in the Kolmogorov turbulence spectrum, has a sound physical basis (for a discussion of using the Kolmogorov model for synthesizing velocity fields see e.g. Stam and Fiume [1993], and for time modulating noise textures see Neyret [2003]).

We finally note that for added realism, our velocity field should vary in time. This is achieved very simply by using time-varying noise. While we have not yet tried it, it seems profitable to look at the pseudo-advection ideas of FlowNoise [Perlin and Neyret 2001] to give more realism to the evolution of the turbulent vortices.

2.3 Modulation

We are not restricted to linear superposition of potentials: thanks to the curl identity, we may manipulate ψ as we like and still be sure of getting an incompressible velocity field. The simplest of such controls is to spatially modulate the flow: e.g. letting the field decay to zero away from objects in the scene, or changing it according to the height in a column of smoke. While simply modulating the velocity field $A(\vec{x})\vec{v}(\vec{x})$ no longer gives a divergence-free field, modulating the potential, $\vec{v} = \nabla \times (A(\vec{x})\psi(\vec{x}))$, does the trick.

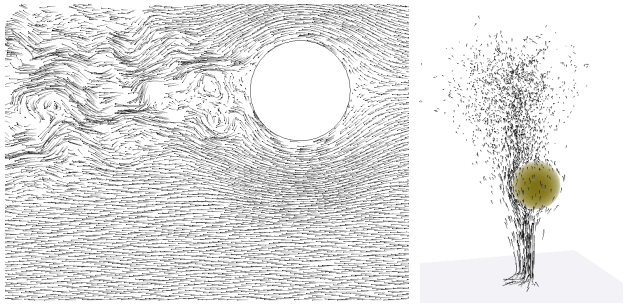

Figure 2: *On the left, we construct a turbulent wake behind a rigid body. On the right, we set up a vortex ring to push fluid past a sphere before diffusing by turbulent mixing. In both cases each octave of turbulence noise is adjusted in a scale-appropriate way to the geometry:* $\psi_T(x) = \sum_i a_i \,\mathrm{ramp}\left(\frac{d(x)}{d_i}\right) N\left(\frac{x}{d_i}, \frac{t}{d_i}\right)$. *We then multiply by a smooth amplitude function to create turbulence only behind the obstacles, and add this to the underlying laminar flow potential which is also smoothly ramped to zero at solid boundaries:* $\psi(x,t) = A(x)\,\psi_T(x,t) + \mathrm{ramp}\left(\frac{d(x)}{d_L}\right)\,\psi_L(x)$. *(In 3D, only tangential components of vector noise are ramped down; see equation 5).*

2.4 Boundaries

Consider a motionless solid object in the flow. The boundary condition viscous flow must satisfy is $\vec{v} = 0$. This can be achieved simply by modulating the potential down to zero with a smoothed step function of distance, so that all the partial derivatives (and hence the curl) of the new potential are zero at the boundary.

Of more interest in animation is the inviscid boundary condition, $\vec{v} \cdot \vec{n} = 0$, requiring that the component of velocity normal to the boundary is zero—allowing the fluid to slip past tangentially but not to flow through a solid. Most turbulent fluids have such small viscosities that this is a more reasonable approximation.

In two dimensions, note that our velocity field is just the 90° rotation of the gradient $\nabla\psi$: if we want the velocity field to be tangent to the boundary, we need the gradient to be perpendicular to the boundary. This happens precisely when the boundary is an isocontour of ψ, i.e. when ψ has some constant value along the boundary. We can achieve this without eliminating the gradient altogether by modulating ψ with a ramp through zero based on distance to the closest boundary point:

$$\psi_{constrained}(\vec{x}) = \mathrm{ramp}\left(\frac{d(\vec{x})}{d_0}\right)\psi(\vec{x}) \quad (3)$$

where $d(\vec{x})$ is the distance to all solid boundaries and d_0 is the width of the modified region—when using noise with length scale L, it makes sense to set $d_0 = L$. We use the following smooth ramp:

$$\mathrm{ramp}(r) = \begin{cases} 1 & : \quad r \geq 1 \\ \frac{15}{8}r - \frac{10}{8}r^3 + \frac{3}{8}r^5 & : \quad 1 > r > -1 \\ -1 & : \quad r \leq -1 \end{cases} \quad (4)$$

In three dimensions things are a little more complicated. Letting $\alpha = |\mathrm{ramp}(d(\vec{x})/d_0)|$ and $\hat{n}$ be the normal to the boundary at the closest point to $\vec{x}$, we use

$$\vec{\psi}_{constrained}(\vec{x}) = \alpha\vec{\psi}(\vec{x}) + (1-\alpha)\hat{n}(\hat{n} \cdot \vec{\psi}(\vec{x})). \quad (5)$$

That is, we ramp down the tangential component of $\vec{\psi}$ near the boundary, but leave the normal component unchanged. This can be proven to give a tangential velocity field at smooth object boundaries, using the fact that $\hat{n}$ is the gradient of signed distance from the boundary (hence its curl vanishes).

Unfortunately, the normal field may be discontinuous along the medial axis of the geometry: naïvely using equation (5) can result in spikes when we take the curl, particularly near sharp edges. This isn't a problem for equation (3) since the distance field is Lipschitz continuous, which is adequate for our purposes. Thus within distance d_0 of edges flagged as sharp in the system we default to (3), i.e. drop the normal component. In the future we plan to investigate more sophisticated solutions.

2.5 Other Potentials

So far we have only constructed noise that respects unmoving solid boundaries. While of course we can superimpose existing flow primitives on the velocity field for richer capabilities, we can also do more with the potential itself.

It is simple to derive a potential corresponding to a rigid body motion with linear velocity $\vec{V}$ and angular velocity $\vec{\omega}$:

$$\vec{\psi}_{rigid}(\vec{x}) = \vec{V} \times (\vec{x} - \vec{x}_0) + \frac{R^2 - ||\vec{x} - \vec{x}_0||^2}{2} \vec{\omega} \tag{6}$$

where $\vec{x}_0$ is an arbitrary reference point and R is an arbitrary reference level. (Note that there are always infinitely many potentials corresponding to the same $\vec{v}$: any two potentials which differ by only the gradient of some scalar field have exactly the same curl.)

Suppose we have a potential $\vec{\psi}$ we wish to modify to respect boundary conditions on a moving rigid object. First we use equation (5) to zero out the normal component of velocity on the object's surface, giving $\vec{\psi}_0$. Then we use equation (6) with x_0 chosen, say, as the center of the rigid body, smoothly blend it to zero away from the object (choosing R to be the radius of the blend region, so that the blended rotational term drops monotonically to zero), and add it to get $\vec{\psi}(\vec{x}) = \vec{\psi}_0 + A(\vec{x})\vec{\psi}_{rigid}(\vec{x})$. We have experimented, for example, with a blending function $A(\vec{x})$ based on inverse squared distance to each rigid body. Note that at the boundary of the rigid object, the velocity is the sum of the rigid motion and a vector field tangent to the surface: by construction this respects the inviscid boundary condition.

While for a single body the reference point is arbitrary, if we have multiple bodies it can help to use the same reference point to make the blend better. In particular, if all bodies are moving with the same rigid motion, then of course we want their potentials to match up to make the blend perfect.

Flow fields around deforming bodies are trickier. For a closed deforming surface, it may even be impossible—if the body doesn't conserve its volume—and thus we leave that for future work.

Some vortex primitives include a simple vortex particle at $\vec{x}_0$ with angular velocity $\vec{\omega}$, radius R, and smooth fall-off function f:

$$\vec{\psi}_{vort}(\vec{x}) = f\left(\frac{||\vec{x} - \vec{x}_0||}{R}\right) \frac{R^2 - ||\vec{x} - \vec{x}_0||^2}{2} \vec{\omega}, \tag{7}$$

and a simple vortex curve, useful for smoke rings and plumes:

$$\vec{\psi}_{curve}(\vec{x}) = f\left(\frac{||\vec{x} - \vec{x}_C||}{R}\right) \frac{R^2 - ||\vec{x} - \vec{x}_C||^2}{2} \vec{\omega}_C \tag{8}$$

where $\vec{x}_C$ is the closest point on the curve to $\vec{x}$, and $\vec{\omega}_C$ is the angular velocity (tangent to the curve). Other interesting flow structures can be similarly created with the rigid motion formulas in mind.

3 Conclusion

In figures 1 and 2 we show some screenshots of simple turbulence examples constructed with the basic formulas in this paper, running in real-time; see `http://www.cs.ubc.ca/~rbridson/` for animations, full details and code. In conclusion, using a simple vector calculus identity, we have unlocked a new procedural toolbox for creating fluid-like velocity fields, including turbulent motion and flow around arbitrary rigid objects.

Acknowledgments

This work was in part supported by a grant from the Natural Sciences and Engineering Research Council of Canada. We also thank the anonymous reviewers for their considerable help in preparing the work for publication.

References

COOK, R. L., AND DEROSE, T. 2005. Wavelet noise. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3, 803–811.

KNISS, 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.

LAMORLETTE, A., AND FOSTER, N. 2002. Structural modeling of flames for a production environment. In *Proc. ACM SIGGRAPH*, 729–735.

NEYRET, F. 2003. Advected textures. In *Proc. Symp. Comp. Anim.*, 147–153.

PATEL, M., AND TAYLOR, N. 2005. Simple divergence-free fields for artistic simulation. *journal of graphics tools 10*, 4, 49–60.

PERLIN, K., AND NEYRET, F. 2001. Flow noise. In *ACM SIGGRAPH Technical Sketches and Applications*, 187. http://www-evasion.imag.fr/Publications/2001/PN01/.

PERLIN, K. 1985. An image synthesizer. In *Proc. ACM SIGGRAPH*, 287–296.

PERLIN, K. 2002. Improving noise. *ACM Trans. Graph. (Proc. SIGGRAPH) 21*, 3, 681–682.

SHINYA, M., AND FOURNIER, A. 1992. Stochastic motion: Motion under the influence of wind. In *Proc. Eurographics*, 119–128.

SIMS, K. 1990. Particle animation and rendering using data parallel computation. In *Proc. ACM SIGGRAPH*, 405–413.

STAM, J., AND FIUME, E. 1993. Turbulent wind fields for gaseous phenomena. In *Proc. ACM SIGGRAPH*, 369–376.

STAM, J. 1997. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum (Proc. Eurographics) 16*, 3, C159–C164.

VON FUNCK, W., THEISEL, H., AND SEIDEL, H.-P. 2006. Vector field based shape deformations. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3, 1118–1125.

WEJCHERT, J., AND HAUMANN, D. 1991. Animation aerodynamics. In *Proc. ACM SIGGRAPH*, 19–22.