

# An HSV representation of non-Newtonian, lattice-Boltzmann flows

Robert Geist, Karl Rasche, and James Westall

Clemson University, Clemson, South Carolina, USA

## ABSTRACT

An approach to modeling and visualization of non-Newtonian flows that is based on lattice-Boltzmann techniques is described. Important advantages over traditional, finite-element methods include the speed and simplicity of the update and the reduced storage, which is linear in the number of nodes. Because the quantities of interest in the solution depend critically upon directional flow densities, a technique for their direct display is suggested. The HSV (hue, saturation, value) color model is a hexcone whose intersection with any plane of constant  $V > 0$  yields a hexagon that matches the isotropic flow grid. Thus, a natural representation emerges. An auxiliary display is provided in which the HSV cone is shown in partial transparency along with the magnitudes of the contributing densities for the pixel currently identified by the cursor. A variation on LIC is suggested to enhance field visualization.

**Keywords:** HSV color model, non-Newtonian flows, lattice-Boltzmann models, LIC noise, polymer melt

## 1. INTRODUCTION

Synthetic fibers have become an integral part of daily life in most of the industrialized world. Each year, U.S. consumers spend more than \$280 billion dollars on apparel, most of it containing synthetic fibers. Extrusion of polymer melt is an essential process in the creation of such fibers. Physical properties of the extrusion devices determine, in non-trivial ways, the extent of mixing of polymer components and additives as well as the residence time of the polymers, at elevated temperatures, within the devices. The extent of component mixing and residence time have a profound effect upon the physical properties, e.g., strength, extensibility, toughness, modulus, of the fibers produced. In a standard industrial extrusion device, chips of solid polymer are loaded into an input hopper, heated to melt in the initial component of the device, and then forced by pressure through small openings in a die or spinnerette attached to the end of the device.

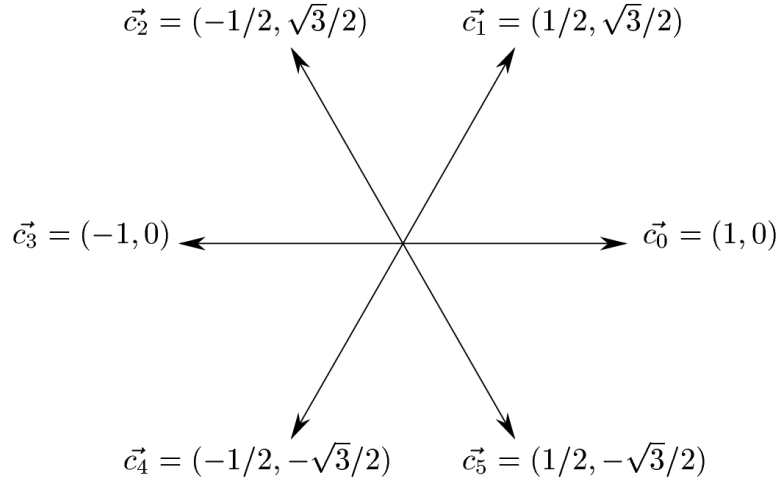
Modeling and visualization of melt flow through an extruder that includes simultaneous representation of polymer temperature, pressure, velocity, and stress can provide valuable keys to predicting fiber properties. Unfortunately, this is a highly non-trivial task. Polymer melt is almost always a non-Newtonian fluid, that is, its viscosity is non-constant and at least dependent upon shear stress. Flow is described, then, by a continuity equation (conservation of mass), a generalized Navier-Stokes equation, and one or more constitutive equations that represent the viscous or viscoelastic behavior. Solution of this system of PDEs is most often achieved by finite-element techniques.

Nevertheless, the geometries of interest are fairly intricate, and accurate representation of flow vortices is crucial to estimating the distribution of residence time. This leads to geometry meshes of high detail and hence many vertices or nodes, often approaching a million or more. Although sparse representation techniques are helpful, this size can be daunting to standard finite-element solvers.

This paper describes an approach to modeling and visualization of non-Newtonian flows through intricate geometries that is based on lattice-Boltzmann techniques, which have recently drawn considerable attention in the literature.<sup>1-3</sup> The lattice-Boltzmann approach maintains, for each node in the geometry at each time instant, a collection of directional densities. The densities are updated, based on neighboring densities, in a synchronous update step according to a relatively simple collision operator that is linearized about a local equilibrium.

---

Further author information: Department of Computer Science, Clemson University, Clemson, SC, USA 29634-0974; email: {rmg|rkarl|westall}@cs.clemson.edu



**Figure 1:** Grid structure and directions.

Typical quantities of interest, including density, velocity field, and momentum tensor, are determined directly from the directional densities. Important advantages over traditional finite-element methods include the speed and simplicity of the update and the reduced storage, which is linear in the number of nodes. Updates are also highly parallelizable, and so a number of processors on the order of the number of geometry nodes may be brought to bear. A geometry with a million or more nodes is not out of reach.

The approach taken here is restricted to two-dimensional flows and is similar to that of Aharonov and Rothmann,<sup>1</sup> in that a hexagonal meshing geometry is selected to ensure isotropic flow, and non-constant viscosity is captured through a dynamically varying collision matrix. However, unlike the model of Ref. 1, a local equilibrium is selected by forcing Galilean invariance, rather than employing the more conventional choice from discrete dynamics due to Frisch, Hasslacher and Pomeau,<sup>4</sup> and the eigenvalues of the collision matrix are selected to allow maximum range for the dynamic viscosity. The derivation of the Navier-Stokes equation is similar to that of Chopard and Droz.<sup>5</sup> However, a hexagonal grid, rather than a weighted rectangular grid, is employed throughout, and some unusual calculations involving Kronecker deltas in Ref. 5 are avoided completely.

Because the quantities of interest in the solution depend critically upon the directional densities, a technique for their direct display is suggested. In particular, the HSV (hue, saturation, value) color model,<sup>6</sup> often used by artists for its intuitive appeal, is a hexcone whose intersection with any plane of constant  $V > 0$  yields a hexagon that matches the isotropic flow grid. Thus, a natural representation for directional densities emerges, where directions are identified with the additive (R,G,B) and subtractive (C,M,Y) primaries. In this representation,  $V$  = density, and (S,H) represents the velocity field in polar coordinates.

The remainder of the paper is organized as follows. In Sec. 2 we describe the structure of the model, the fundamental update equation, and the relationship between model parameters and standard measures of interest in flow visualization. In Sec. 3, we show how the simple update equation leads directly to the continuity equation for flows, the Euler equation of hydrodynamics, and the Navier-Stokes equation. In Sec. 4, we consider the density collision operator and its relation to so-called *power-law fluids*. In Sec. 5, we derive the Galilean invariant local equilibrium and show that its use yields the more conventional form of the Navier-Stokes equation. The HSV representation is provided in Sec. 6, and the augmented representation using line integral convolution (LIC) filtering is provided in Sec. 7. Conclusions follow in Sec. 8.

## 2. MODEL STRUCTURE

Lattice-Boltzmann models are state-based models with update rules that are applied in a synchronous fashion. Although rectangular lattices are computationally convenient, they impose difficulties in achieving isotropic flow behavior. Therefore, we start with a hexagonal lattice of nodes in the plane as shown in Fig. 1. We assume a

lattice spacing (the distance to the next node),  $\lambda$ , a time step,  $\tau$ , unit velocity  $v = (\lambda/\tau)$ , and velocity vectors  $\vec{v}_i = v\vec{c}_i$ ,  $i = 0, \dots, 5$ . The key quantity of interest is the directional density,  $f_i(\vec{r}, t)$ , which is the particle density at lattice location  $\vec{r}$  at time  $t$  moving in direction  $\vec{c}_i$ . The general case will have  $f_i(\cdot) \in [0, 1]$ . Quantities of interest related to the  $f_i(\cdot)$ s are:

- density,  $\rho(\vec{r}, t) = \sum_{i=0}^5 f_i(\vec{r}, t)$
- velocity field,  $\vec{u}(\vec{r}, t) = (\sum_{i=0}^5 \vec{v}_i f_i(\vec{r}, t)) / (\sum_{i=0}^5 f_i(\vec{r}, t))$
- momentum tensor,  $\Pi_{\alpha\beta} = \sum_{i=0}^5 v_{i\alpha} v_{i\beta} f_i(\vec{r}, t)$ , where  $\alpha, \beta \in \{x, y\}$

The fundamental system update equation (basis for simulation) is given by:

$$f_i(\vec{r} + \lambda\vec{c}_i, t + \tau) = f_i(\vec{r}, t) + \Theta_i(f(\vec{r}, t)), \quad i = 0, 1, \dots, 5 \quad (1)$$

where  $\Theta : \mathfrak{R}^6 \rightarrow \mathfrak{R}^6$  is a *collision operator*, to be specified. Assuming that  $\Theta$  is easy to compute, we have in Eqn. (1) a synchronous update that is fast and parallelizable. It remains to be seen how this describes fluid flow, but some important properties of  $\Theta$  can be specified immediately. From Eqn. (1) we have

$$f_i(\vec{r} + \lambda\vec{c}_i, t + \tau) - f_i(\vec{r}, t) = \Theta_i(f(\vec{r}, t)), \quad i = 0, 1, \dots, 5 \quad (2)$$

and so we can express:

- conservation of mass:  $\sum_{i=0}^5 \Theta_i(f(\vec{r}, t)) = 0$
- conservation of momentum:  $\sum_{i=0}^5 \vec{v}_i \Theta_i(f(\vec{r}, t)) = (0, 0)$

### 3. NAVIER-STOKES

We now verify that the fundamental update equation, Eqn. (2), describes fluid flow by deriving the Navier-Stokes equation directly from it. Although the notation is a bit tedious, the procedure is straightforward. The primary tool of the derivation is a dimension 3 Taylor expansion. Recall the form (see Ref.7):

$$f(x + h, y + k, t + l) = f(x, y, t) + [(h, k, l) \cdot \nabla] f(x, y, t) + \frac{[(h, k, l) \cdot \nabla]^2}{2!} f(x, y, t) + \dots \quad (3)$$

where the square term is not the Laplacian, in that it includes cross terms. If we apply this to the basic update Eqn. (2) we get:

$$[(\lambda\vec{c}_i, \tau) \cdot \nabla] f_i(\vec{r}, t) + \frac{[(\lambda\vec{c}_i, \tau) \cdot \nabla]^2}{2!} f_i(\vec{r}, t) + \dots = \Theta_i(f(\vec{r}, t)) \quad (4)$$

We want to consider the limiting behavior here as  $\lambda, \tau \rightarrow 0$ ; they can, of course, approach at different rates, and it turns out that two sets of rates are important. Write

$$t = \frac{t_1}{2\epsilon} + \frac{t_2}{2\epsilon^2} \quad \text{where} \quad t_1 = o(\epsilon), t_2 = o(\epsilon^2)$$

$$\vec{r} = \frac{\vec{r}_1}{\epsilon} \quad \text{where} \quad \vec{r}_1 = o(\epsilon)$$

Then

$$\begin{aligned} \frac{\partial}{\partial t} &= \epsilon \frac{\partial}{\partial t_1} + \epsilon^2 \frac{\partial}{\partial t_2} \\ \frac{\partial}{\partial r_\alpha} &= \epsilon \frac{\partial}{\partial r_{1\alpha}} \quad \text{for} \quad \alpha \in \{x, y\} \end{aligned}$$

So

$$\nabla = (\partial/\partial r_x, \partial/\partial r_y, \partial/\partial t) = (\epsilon\partial/\partial r_{1x}, \epsilon\partial/\partial r_{1y}, \epsilon\partial/\partial t_1 + \epsilon^2\partial/\partial t_2) \quad (5)$$

We also assume that the solution,  $f(\vec{r}, t)$ , is a small perturbation on this same scale about some local equilibrium, i.e.,

$$f(\vec{r}, t) = f^0(\vec{r}, t) + \epsilon f^1(\vec{r}, t) \quad (6)$$

and that the macroscopic quantities are carried by the equilibrium value, that is,

$$\rho(\vec{r}, t) = \sum_{i=0}^5 f_i^0(\vec{r}, t) \quad (7)$$

$$\rho(\vec{r}, t)\vec{u}(\vec{r}, t) = \sum_{i=0}^5 \vec{v}_i f_i^0(\vec{r}, t) \quad (8)$$

Using the gradient expression in Eqn. (5) and the local equilibrium expression in Eqn. (6), we can sum Eqn. (4) over  $i = 0, 1, \dots, 5$ , divide by  $\tau$  (recall that  $\vec{v}_i = \lambda \vec{c}_i / \tau$ ), and equate coefficients of  $\epsilon^1$  to obtain

$$\text{div}_1[\rho(\vec{r}, t)\vec{u}(\vec{r}, t)] + \partial\rho(\vec{r}, t)/\partial t_1 = 0, \quad (9)$$

where the right hand side disappears by conservation of mass. This is the *continuity equation* for time scale  $t_1$ .

The remainder of the derivation is a repetition of this fundamental step, that is, we equate coefficients of various powers of  $\epsilon$  in Eqn. (4). As a first instance, if we multiply Eqn. (4) by  $\vec{v}_i = (v_{ix}, v_{iy})$ , sum over  $i = 0, 1, \dots, 5$ , divide by  $\tau$ , and again equate coefficients of  $\epsilon^1$ , we obtain a pair of equations:

$$\frac{\partial}{\partial t_1}[\rho(\vec{r}, t)\vec{u}(\vec{r}, t)]_\alpha + \frac{\partial}{\partial r_{1x}}(\Pi^0(\vec{r}, t))_{\alpha x} + \frac{\partial}{\partial r_{1y}}(\Pi^0(\vec{r}, t))_{\alpha y} = 0 \quad \text{for } \alpha \in \{x, y\} \quad (10)$$

where the right hand side vanishes due to conservation of momentum, and  $\Pi^0$  denotes the momentum tensor based on the local equilibrium,  $f^0$ . This is the *Euler Equation of Hydrodynamics*, which is just the Navier-Stokes equation without the dissipative effects of viscosity.

We can repeat the procedure for the coefficients of  $\epsilon^2$ . Because the macroscopic quantities are carried by  $f^0$ , we know that  $\sum_{i=0}^5 f_i^1(\vec{r}, t) = 0$  and  $\sum_{i=0}^5 \vec{v}_i f_i^1(\vec{r}, t) = 0$ . Therefore, if we sum Eqn. (4) over  $i = 0, 1, \dots, 5$ , divide by  $\tau$ , and equate coefficients of  $\epsilon^2$  we obtain

$$\frac{\partial}{\partial t_2}\rho(\vec{r}, t) + (\tau/2)\frac{\partial^2}{\partial t_1^2}\rho(\vec{r}, t) + (\tau/2) \sum_{\alpha, \beta \in \{x, y\}} \frac{\partial^2}{\partial r_{1\alpha}\partial r_{1\beta}}\Pi_{\alpha\beta}^0 + \tau \sum_{\alpha \in \{x, y\}} \frac{\partial^2}{\partial r_{1\alpha}\partial t_1}\rho(\vec{r}, t)\vec{u}(\vec{r}, t)_\alpha = 0 \quad (11)$$

But, by the order 1 continuity equation and the Euler equation, this reduces to

$$\frac{\partial}{\partial t_2}\rho(\vec{r}, t) = 0, \quad (12)$$

that is, density does not change at this time scale. If we now multiply the order 1 continuity equation, Eqn. (9), by  $\epsilon$ , multiply Eqn. (12) by  $\epsilon^2$ , and add we obtain

$$\text{div}[\rho\vec{u}] + \frac{\partial}{\partial t}\rho = 0 \quad (13)$$

which is the standard continuity equation.

The remaining step in this sequence is clear: we multiply both sides of Eqn. (4) by  $\vec{v}_i = (v_{ix}, v_{iy})$ , sum over  $i = 0, 1, \dots, 5$ , divide by  $\tau$ , and equate coefficients of  $\epsilon^2$ . Because the notation is tedious, we restrict to the  $x$  component. Collecting terms, we have

$$\begin{aligned} & \frac{\partial}{\partial t_2}[\rho(\vec{r}, t)\vec{u}(\vec{r}, t)_x] + \sum_{\alpha \in \{x, y\}} \frac{\partial}{\partial r_{1\alpha}}[\Pi^1(\vec{r}, t)]_{\alpha x} + (\tau/2)\frac{\partial^2}{\partial t_1^2}[\rho(\vec{r}, t)\vec{u}(\vec{r}, t)_x] + \\ & \tau \frac{\partial}{\partial t_1} \sum_{\alpha \in \{x, y\}} \frac{\partial}{\partial r_{1\alpha}}[\Pi^0(\vec{r}, t)]_{\alpha x} + (\tau/2) \sum_{\alpha, \beta \in \{x, y\}} \frac{\partial^2}{\partial r_{1\alpha}\partial r_{1\beta}}[S^0(\vec{r}, t)]_{\alpha\beta x} = 0 \end{aligned} \quad (14)$$

where  $S^0$  is a third-order tensor,

$$S^0(\vec{r}, t)_{\alpha\beta\gamma} = \sum_{i=0}^5 v_{i\alpha} v_{i\beta} v_{i\gamma} f_i^0(\vec{r}, t)$$

From the Euler equation, the third and fourth summands in (14) can now be collapsed, and we are left with:

$$\frac{\partial}{\partial t_2} [\rho(\vec{r}, t) \vec{u}(\vec{r}, t)_x] + \sum_{\alpha \in \{x, y\}} \frac{\partial}{\partial r_{1\alpha}} \left[ \Pi^1(\vec{r}, t)_{\alpha x} + (\tau/2) \left( \frac{\partial}{\partial t_1} \Pi^0(\vec{r}, t)_{\alpha x} + \sum_{\beta \in \{x, y\}} \frac{\partial}{\partial r_{1\beta}} [S^0(\vec{r}, t)]_{\alpha\beta x} \right) \right] = 0 \quad (15)$$

This contains the dissipative contributions to the flow. Multiplying the Euler equation by  $\epsilon$ , multiplying the dissipative contributions, Eqn. (15), by  $\epsilon^2$ , and adding, we obtain

$$\frac{\partial}{\partial t} [\rho(\vec{r}, t) \vec{u}(\vec{r}, t)_x] + \sum_{\alpha \in \{x, y\}} \frac{\partial}{\partial r_\alpha} \left[ \Pi(\vec{r}, t)_{\alpha x} + (\tau/2) \left( \epsilon \frac{\partial}{\partial t_1} \Pi^0(\vec{r}, t)_{\alpha x} + \sum_{\beta \in \{x, y\}} \frac{\partial}{\partial r_\beta} [S^0(\vec{r}, t)]_{\alpha\beta x} \right) \right] = 0 \quad (16)$$

This is the Navier-Stokes equation, although we still need to express  $\Pi$ ,  $\Pi^0$  and  $S^0$  in terms of  $\rho$  and  $\vec{u}$ . This will require specification of the collision operator,  $\Theta$ . It is worth noting that, to this point, constraints on the specification of the collision operator are minimal.

#### 4. POWER-LAW FLUIDS

The first step in the specification of the collision operator,  $\Theta$ , is to expand about  $f^0$ . Recalling Eqn. (6), we have:

$$\Theta(f(\vec{r}, t)) = \Theta(f^0(\vec{r}, t)) + \left[ \frac{\partial \Theta_i}{\partial f_j} \right]_{f=f^0} (\epsilon f^1) + O(\epsilon^2) \quad (17)$$

Let

$$\Omega_{i,j} = \left[ \frac{\partial \Theta_i}{\partial f_j} \right]_{f=f^0}$$

If we now equate powers of  $\epsilon^0$  in Eqn. (4) we obtain

$$\Theta(f^0(\vec{r}, t)) = 0$$

that is, the collision operator vanishes at the local equilibrium.

The fundamental update equation, Eqn. (1), now reads:

$$f_i(\vec{r} + \lambda \vec{c}_i, t + \tau) = f_i(\vec{r}, t) + \Omega_i(f(\vec{r}, t) - f^0(\vec{r}, t)) \quad (18)$$

where  $\Omega_i$  denotes the  $i^{th}$  row of matrix  $\Omega$ . We can use conservation of mass and conservation of momentum to deduce the structure of  $\Omega$ . From conservation of mass we have:

$$\begin{aligned} \sum_{i=0}^5 \Omega_{i,j} &= \sum_{i=0}^5 \frac{\partial \Theta_i}{\partial f_j} \\ &= \frac{\partial}{\partial f_j} \sum_{i=0}^5 \Theta_i \\ &= 0 \end{aligned}$$

that is, the column sums are 0. Similarly, from conservation of momentum, we have:

$$\begin{aligned} \sum_{i=0}^5 \vec{v}_i \Omega_{i,j} &= \frac{\partial}{\partial f_j} \sum_{i=0}^5 \vec{v}_i \Theta_i \\ &= (0, 0) \end{aligned}$$

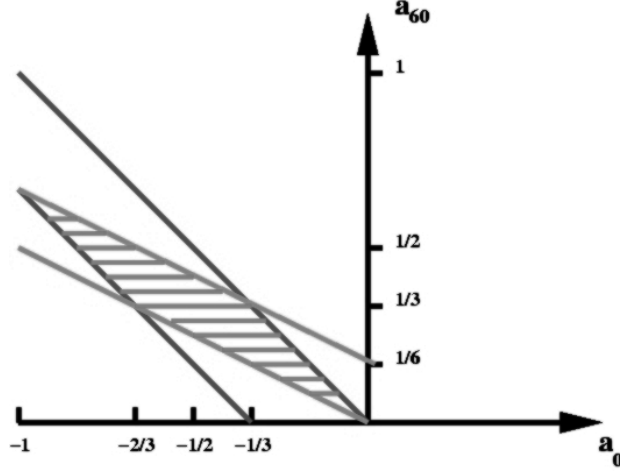


Figure 2:  $e_1$  and  $e_2$  restrictions

and so weighted column sums are also 0.

More generally,  $\Omega_{i,j}$  is the deflection of density  $f_i$  in the  $j^{\text{th}}$  direction. Isotropic flow dictates that its value depends only upon the angle between  $i$  and  $j$ , which, for our hex-grid, can be only  $0^\circ, 60^\circ, 120^\circ$ , or  $180^\circ$ . Using the notation of Higuera, Succi, and Benzi,<sup>8</sup> we must then have:

$$\Omega = \begin{pmatrix} a_0 & a_{60} & a_{120} & a_{180} & a_{120} & a_{60} \\ a_{60} & a_0 & a_{60} & a_{120} & a_{180} & a_{120} \\ a_{120} & a_{60} & a_0 & a_{60} & a_{120} & a_{180} \\ a_{180} & a_{120} & a_{60} & a_0 & a_{60} & a_{120} \\ a_{120} & a_{180} & a_{120} & a_{60} & a_0 & a_{60} \\ a_{60} & a_{120} & a_{180} & a_{120} & a_{60} & a_0 \end{pmatrix} \quad (19)$$

The matrix is symmetric (and circulant), and so the results for column sums apply to row sums. In particular, 0 is a triple eigenvalue with eigenvectors  $(1, 1, 1, 1, 1, 1)$ ,  $(v_{0x}, v_{1x}, v_{2x}, v_{3x}, v_{4x}, v_{5x})$ , and  $(v_{0y}, v_{1y}, v_{2y}, v_{3y}, v_{4y}, v_{5y})$ . Equivalently, we can use  $(1, 1, 1, 1, 1, 1)$ , the  $x$  components of the  $\vec{c}_i$ s,  $c_x = (1, 1/2, -1/2, -1, -1/2, 1/2)$ , and the  $y$  components of the  $\vec{c}_i$ s,  $c_y = (0, \sqrt{3}/2, \sqrt{3}/2, 0, -\sqrt{3}/2, -\sqrt{3}/2)$  as the eigenvectors for 0.

It is easy to verify that there are two non-zero eigenvalues,  $e_1 = 6a_0 + 6a_{60}$ , and  $e_2 = -6a_0 - 12a_{60}$ .  $e_1$  is a double value with eigenvectors  $V_0 = (1, 0, -1, 1, 0, -1)$  and  $V_1 = (1, -2, 1, 1, -2, 1)$ . The eigenvector for  $e_2$  is  $V_2 = (1, -1, 1, -1, 1, -1)$ . This leaves a great deal of flexibility in the choice of  $\Omega$ , but stability requires that the eigenvalues are  $\in (-2, 0)$ . (Subtract  $f^0$  from both sides of the update equation; the eigenvalues of  $I + \Omega$  must be  $\in (-1, 1)$ .)

A further restriction is the target viscosity,  $\mu$ , which we will see (in section 5) is related to the eigenvalue  $e_1$  according to:

$$\mu = -\frac{\tau v^2}{4} \left( \frac{1}{e_1} + \frac{1}{2} \right) \quad (20)$$

In Fig. 2 we plot the inequalities  $e_1, e_2 \in (-2, 0)$  on  $a_0/a_{60}$  axes. A specification of viscosity,  $\mu \in (0, +\infty)$ , amounts to selection of a line segment in Fig. 2 that is parallel to the bounding lines  $a_{60} = -a_0$  and  $a_{60} = -a_0 - 1/3$ . Note that a choice of  $a_{60} = 1/3$  is convenient in that it is available on all such line segments. With this choice, viscosity is then related to  $a_0$  by:

$$a_0 = -\frac{1}{3} \left( \frac{8\mu + 2\tau v^2}{8\mu + \tau v^2} \right) \quad (21)$$

Although we have yet to specify  $f^0$ , it is now easy to see how to simulate non-Newtonian, power-law fluids. The defining characteristic of such fluids is that viscosity depends upon strain rate according to an equation of

the form:

$$\mu = K(\dot{\epsilon})^n \quad (22)$$

where  $\dot{\epsilon}$  denotes the so-called second invariant of the rate of deformation tensor, specifically,

$$\dot{\epsilon} = \sqrt{\left(\frac{\partial u_x}{\partial x}\right)^2 + \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right)^2 / 2 + \left(\frac{\partial u_y}{\partial y}\right)^2} \quad (23)$$

In this form, the choices,  $n = 0$ ,  $n < 0$ , and  $n > 0$ , denote Newtonian, strain-thinning, and strain-thickening fluids.

Given  $K$  and  $n$  that specify the power-law fluid, we can simulate the flow rather easily. For each node:

1. Use numerical estimates of velocity derivatives to compute  $\dot{\epsilon}$  by Eqn. (23).
2. Use Eqn. (22) to compute target viscosity,  $\mu$ .
3. Use Eqn. (21) to compute  $a_0$ .
4. Complete specification of  $\Omega$  by:  $a_{60} = 1/3$ ,  $a_{120} = -(2a_0 + 3a_{60})$ ,  $a_{180} = (3a_0 + 4a_{60})$ .
5. Update  $f()$  for this node by Eqn. (18).

## 5. NAVIER-STOKES REVISITED

To obtain the standard form of the Navier-Stokes equation, we yet need to specify the local equilibrium,  $f^0$ . An elegant derivation of  $f^0$  appears in Ref.,<sup>3</sup> where it is observed that the choice of  $f^0$  is not unique and that the particular value selected can affect the convergence rate of the update scheme. Nevertheless, when that technique is applied to our 6 population, hex-grid model, it yields an overly-constrained system with no (non-trivial) solution. Karlin (personal communication) believes that his 9 population, rectangular grid model is the minimum 2D system for which his technique will yield solutions.

As an alternative, we make the simplifying assumption that  $f^0$  is quadratic in the velocity components,  $u_x$  and  $u_y$ , so that

$$f_i^0 = A_i + B_i u_x + C_i u_y + D_i u_x u_y + E_i u_x^2 + G_i u_y^2 \quad (24)$$

The constraints on  $f^0$  are given by Eqs. (7)-(8) together with a constraint on the momentum tensor:

$$\sum_i v_{ix} v_{iy} f_i^0 = \rho(u_x u_y + \delta_{xy} c_s^2) \quad (25)$$

where  $c_s$  denotes the speed of sound (to be determined), and  $\delta_{xy}$  denotes Kronecker delta. The constraint on the momentum tensor will be required in order to obtain the standard Navier-Stokes equation from our current form, Eqn. (16).

If we are willing to assume that the directional coefficients in Eqn. (24) are proportional to their velocities,  $\vec{v}_i$ , we can obtain a unique solution. Specifically, if we write:

$$(A_i, B_i, C_i, D_i, E_i) = (A, v_{ix} B, v_{iy} C, v_{ix} v_{iy} D, (v_{ix}^2 - v^2/2)E, (v_{iy}^2 - v^2/2)G) \quad (26)$$

then we can start with the constraints of Eqs. (7), (8), and (25) and apply the following (arithmetic) identities:

$$\sum v_{i\alpha} = 0 \quad \alpha \in \{x, y\} \quad (27)$$

$$\sum v_{i\alpha}^2 = 3v^2 \quad \alpha \in \{x, y\} \quad (28)$$

$$\sum v_{ix} v_{iy} = 0 \quad (29)$$

$$\sum v_{i\alpha}^2 v_{i\beta} = 0 \quad \alpha, \beta \in \{x, y\} \quad (30)$$

$$\sum v_{ix}^2 v_{iy}^2 = (3/4)v^4 \quad (31)$$

$$\sum v_{i\alpha}^3 v_{i\beta} = 0 \quad \alpha, \beta \in \{x, y\}, \alpha \neq \beta \quad (32)$$

$$\sum v_{i\alpha}^4 = (9/4)v^4 \quad \alpha \in \{x, y\} \quad (33)$$

to obtain the solution. The final form is:

$$f_i^0 = \frac{\rho}{6} + \frac{\rho}{3v^2} [v_{ix}u_x + v_{iy}u_y] + \frac{4\rho}{3v^4} [v_{ix}v_{iy}u_xu_y] + \frac{2\rho}{3v^4} [(v_{ix}^2 - v^2/2)u_x^2 + (v_{iy}^2 - v^2/2)u_y^2], \quad (34)$$

and  $c_s^2 = (v^2 - u^2)/2$  where  $u^2 = u_x^2 + u_y^2$ .

The remaining task at hand is the development of Eqn. (16) to express  $\Pi$ ,  $\Pi^0$ , and  $S^0$  in terms of  $\rho$  and  $\vec{u}$ . Since  $f^0$  is specified, only the expression for  $\Pi$  will require significant effort. Recall that

$$\Pi_{\alpha,\beta} = \sum_{i=0}^5 v_{i\alpha}v_{i\beta}(f_i^0 + \epsilon f_i^1), \quad (35)$$

and so we really need an expression for  $f_i^1$  in terms of  $\rho$  and  $\vec{u}$ .

We have previously summed Eqn. (4) over  $i = 0, 1, \dots, 5$  and equated coefficients of various powers of  $\epsilon$ . In section 4 we also equated coefficients of  $\epsilon^0$  in (4) directly, that is, without summing. Following this same approach, we now equate coefficients of  $\epsilon^1$  in Eqn. 4 (4) to obtain

$$\tau \left( \sum_{\alpha \in \{x,y\}} v_{i\alpha} \frac{\partial f_i^0}{\partial r_{1\alpha}} + \frac{\partial f_i^0}{\partial t_1} \right) = \Omega_i f^1 \quad (36)$$

For the purposes of differentiating  $f_i^0$ , we regard it as a function of  $\rho$  and  $\rho\vec{u}$  and use a simplified, order 1 (in  $\vec{u}$ ) approximation,

$$f_i^0 = \frac{\rho}{6} + \frac{v_{ix}}{3v^2} \rho u_x + \frac{v_{iy}}{3v^2} \rho u_y \quad (37)$$

For  $\alpha \in \{x, y\}$  a chain-rule differentiation then gives:

$$\frac{\partial f_i^0}{\partial r_{1\alpha}} = \frac{1}{6} \frac{\partial \rho}{\partial r_{1\alpha}} + \sum_{\beta \in \{x,y\}} \frac{v_{i\beta}}{3v^2} \frac{\partial \rho u_\beta}{\partial r_{1\alpha}} \quad (38)$$

and, similarly,

$$\frac{\partial f_i^0}{\partial t_1} = \frac{1}{6} \frac{\partial \rho}{\partial t_1} + \sum_{\beta \in \{x,y\}} \frac{v_{i\beta}}{3v^2} \frac{\partial \rho u_\beta}{\partial t_1} \quad (39)$$

In this last expression, we can change temporal derivatives to spatial ones by using the continuity equation and the Euler equation. We have:

$$\frac{\partial f_i^0}{\partial t_1} = \frac{1}{6} (-div_1(\rho\vec{u})) + \sum_{\beta \in \{x,y\}} \frac{v_{i\beta}}{3v^2} \left[ \sum_{\gamma \in \{x,y\}} -\frac{\partial \Pi_{\beta\gamma}^0}{\partial r_{1\gamma}} \right] \quad (40)$$

To differentiate  $\Pi^0$ , we again use an order 1 approximation. From Eqn. (25) we have  $\Pi_{\alpha\beta}^0 = \rho u_\alpha u_\beta + \rho \delta_{\alpha\beta} c_s^2$ , and so, to order 1,  $\Pi_{\alpha\beta}^0 = \rho \delta_{\alpha\beta} v^2/2$ . Thus

$$\frac{\partial f_i^0}{\partial t_1} = \frac{1}{6} (-div_1(\rho\vec{u})) - \sum_{\beta \in \{x,y\}} \frac{v_{i\beta}}{6} \frac{\partial \rho}{\partial r_{1\beta}} \quad (41)$$

We can now substitute Eqn. (38) and Eqn. (41) into Eqn. (36) to obtain a substantially simplified expression:

$$\Omega_i f^1 = \tau \left[ \frac{1}{3} \sum_{\alpha \in \{x,y\}} \sum_{\beta \in \{x,y\}} (c_{i\alpha} c_{i\beta} - \delta_{\alpha\beta}/2) \frac{\partial \rho u_\alpha}{\partial r_{1\beta}} \right] \quad (42)$$

We cannot obtain an expression for  $f^1$  by inverting  $\Omega$  (recall, 0 is an eigenvalue of  $\Omega!$ ), but the vectors on the right hand side of Eqn. (42),  $\langle c_{i\alpha}c_{i\beta} - \delta_{\alpha\beta}/2 \rangle$ , are easily seen to be orthogonal to each of the three eigenvectors that span the kernel space of  $\Omega$ , and thus they are in the image. In particular, we can write each as a linear combination of the two eigenvectors  $V_0$  and  $V_1$  for eigenvalue  $e_1$ :

$$\begin{aligned}\langle c_{ix}^2 - 1/2 \rangle &= (3/8)V_0 + (1/8)V_1 \\ \langle c_{iy}^2 - 1/2 \rangle &= -(3/8)V_0 - (1/8)V_1 \\ \langle c_{ix}c_{iy} \rangle &= (\sqrt{3}/8)V_0 - (\sqrt{3}/8)V_1\end{aligned}$$

and so we can conclude that

$$f_i^1 = \frac{\tau}{3e_1} \left[ (c_{ix}^2 - 1/2) \frac{\partial \rho u_x}{\partial r_{1x}} + (c_{ix}c_{iy}) \left( \frac{\partial \rho u_x}{\partial r_{1y}} + \frac{\partial \rho u_y}{\partial r_{1x}} \right) + (c_{iy}^2 - 1/2) \frac{\partial \rho u_y}{\partial r_{1y}} \right] \quad (43)$$

We can now return to Eqn. (16) and assemble all the required terms. The collisional viscosity term,  $\Pi_{\alpha x}$ , has a first summand

$$\Pi_{\alpha x}^0 = \rho u_\alpha u_x + \rho \delta_{\alpha x} c_s^2 \quad (44)$$

and so

$$\sum_{\alpha \in \{x,y\}} \frac{\partial \Pi_{\alpha x}^0}{\partial r_\alpha} = \frac{\partial}{\partial r_x} [\rho u_x^2 + \rho c_s^2] + \frac{\partial}{\partial r_y} [\rho u_x u_y] \quad (45)$$

The second summand of the collisional viscosity term is

$$\epsilon \Pi_{\alpha x}^1 = \sum_{i=0}^5 \epsilon f_i^1 v_{i\alpha} v_{ix} \quad (46)$$

If we differentiate Eqn. (46) with respect to  $r_\alpha$  we obtain

$$\begin{aligned}\frac{\tau v^2}{4e_1} \left[ \frac{\partial^2 \rho u_x}{\partial r_x^2} - \frac{\partial^2 \rho u_y}{\partial r_x \partial r_y} \right] & \quad \alpha = x \\ \frac{\tau v^2}{4e_1} \left[ \frac{\partial^2 \rho u_x}{\partial r_y^2} + \frac{\partial^2 \rho u_y}{\partial r_x \partial r_y} \right] & \quad \alpha \neq x\end{aligned}$$

and so their sum gives:

$$\sum_{\alpha \in \{x,y\}} \frac{\partial}{\partial r_\alpha} \epsilon \Pi_{\alpha x}^1 = \frac{\tau v^2}{4e_1} \nabla^2 \rho u_x \quad (47)$$

The two lattice viscosity terms are handled similarly. For the first,

$$\begin{aligned}\frac{\tau \epsilon}{2} \frac{\partial}{\partial t_1} \Pi_{\alpha x}^0 & \approx \frac{\tau \epsilon}{2} \frac{\partial}{\partial t_1} [\rho \delta_{\alpha x} v^2 / 2] \quad \text{order 1} \\ & = \frac{-\tau v^2}{4} \text{div}(\rho \vec{u}) \delta_{\alpha x}\end{aligned} \quad (48)$$

where we have again used the continuity equation for time scale  $t_1$ . Thus

$$\sum_{\alpha \in \{x,y\}} \frac{\partial}{\partial r_\alpha} \frac{\tau \epsilon}{2} \frac{\partial}{\partial t_1} \Pi_{\alpha x}^0 = \frac{-\tau v^2}{4} \frac{\partial}{\partial r_x} \text{div}(\rho \vec{u}) \quad (49)$$

For the second lattice viscosity term, we can use Eqn. (34) and Eqs. (27)-(33), to obtain

$$S_{\alpha\beta x}^0 = \frac{\rho}{3v^2} \sum_{i=0}^5 (v_{i\alpha} v_{i\beta} v_{ix}^2 u_x + v_{i\alpha} v_{i\beta} v_{ix} v_{iy} u_y) \quad (50)$$

and so

$$\frac{\tau}{2} \sum_{\alpha} \sum_{\beta} \frac{\partial}{\partial r_{\alpha}} \frac{\partial}{\partial r_{\beta}} S_{\alpha\beta x}^0 = \frac{\tau v^2}{4} \left[ \frac{\partial}{\partial r_x} \text{div}(\rho \vec{u}) + (1/2) \nabla^2(\rho u_x) \right] \quad (51)$$

Finally, by substituting Eqs. (45),(47),(49), and (51) into the previous version of the Navier-Stokes equation, Eqn. (16), we arrive at:

$$\frac{\partial}{\partial t} \rho u_x + (\vec{u} \cdot \nabla)(\rho u_x) + \text{div}(\vec{u}) \rho u_x = -\frac{\partial}{\partial r_x} \rho c_s^2 - \frac{\tau v^2}{4} \left( \frac{1}{e_1} + \frac{1}{2} \right) \nabla^2(\rho u_x) \quad (52)$$

For incompressible fluids, we have  $\rho = \text{constant}$  and  $\text{div}(\rho \vec{u}) = 0$ , and so we obtain (for the  $x$  component)

$$\frac{\partial}{\partial t} u_x + (\vec{u} \cdot \nabla) u_x = -(1/\rho) \frac{\partial}{\partial r_x} \rho c_s^2 - \frac{\tau v^2}{4} \left( \frac{1}{e_1} + \frac{1}{2} \right) \nabla^2(u_x), \quad (53)$$

and thus for both components:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -(1/\rho) \nabla(\rho c_s^2) - \frac{\tau v^2}{4} \left( \frac{1}{e_1} + \frac{1}{2} \right) \nabla^2(\vec{u}), \quad (54)$$

the classical form. Note the appearance of the viscosity term containing  $e_1$  and  $\tau v^2$ . The latter suggests the need for the two time scales.

In practice, we find that stability of simulations based on Eqn. (1) requires the addition of an inert, rest density at each node. This alters some of the constants derived here, but the steps in the calculations are identical to those presented.

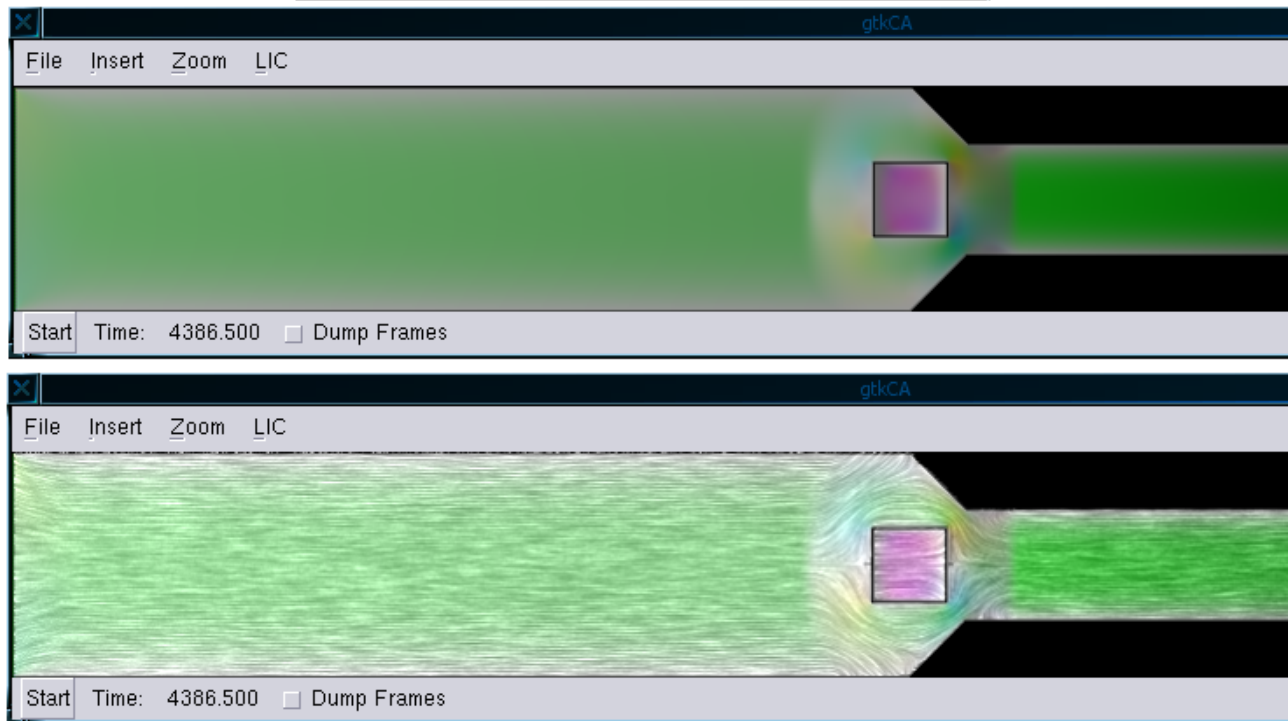
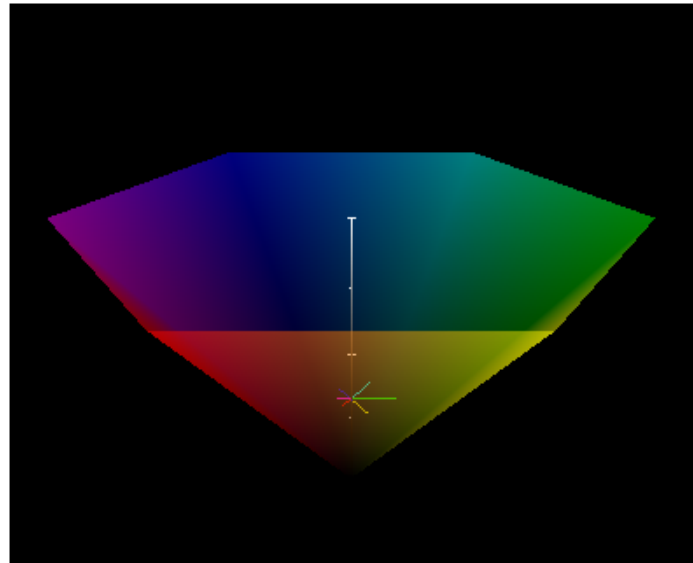
## 6. HSV REPRESENTATION

The hue, saturation, value color model is represented by a hexcone as shown at the top of the color plate, Fig. 3. Value (approximately, intensity) is measured as height along the central axis of the cone. It ranges from black ( $V = 0$ ) to white ( $V = 1$ ). Saturation and hue are  $(r, \theta)$  polar coordinates in the plane orthogonal to the central axis. Additive (red, green, blue) and subtractive (cyan, magenta, yellow) primaries are represented by hexagon vertices in the plane  $V = 1$ .

This structure provides a natural representation for the directional densities in our hexgrid. In particular, we can map density,  $\rho$ , to value and velocity,  $\vec{u}$ , in polar coordinates to (saturation,hue). In the image immediately beneath the cone in Fig. 3 we use this representation to show a snapshot from a simulated flow of a strain-thinning fluid ( $K = 0.1$ ,  $n = -0.5$ ) in a simple channel. The flow is predominantly left to right, as indicated by the green hue. The gray regions near the channel boundaries are low velocity regions caused by the no-slip boundary condition, which is implemented via a reflective collision rule. When a target node of Eqn. (1) is labeled “boundary”, the update density is returned to the source and given opposite direction. The net flow at the boundary is zero. Cross-channel velocities show a flattened quadratic profile, which is characteristic of strain-thinning fluids.

Mapping directional densities to a single pixel reduces information, e.g., if any two opposing directional densities are at 1.0 and all others at 0.0, we obtain  $H = S = 0$ , and the contributing directions are lost. Although the effects of such loss are minor for real flows, an auxiliary display is provided in which the HSV cone is shown in partial transparency along with the magnitudes of the contributing densities for the pixel currently identified by the cursor. This is similar in spirit to the derived space representations suggested by Henze.<sup>9</sup> The cone in Fig. 3 shows an example. The contributing densities are displayed as vectors orthogonal to the central axis.

Our simulation tool also allows interactive manipulation of geometry. Boundaries to flow can be drawn, erased, or moved in real time. The small box in the center of the channel was created just prior to the snapshot. The flow trapped inside has reflected from the right wall and is now moving in the opposite direction, as indicated by the magenta hue. Vortices in the flow, identified by a local rainbow spectrum, can be easily induced by adding wings that emanate from the channel boundaries.



**Figure 3.** HSV cone showing component directional densities (top), example strain-thinning flow (middle), and flow augmented with LIC (bottom).

## 7. LIC STREAMLINES

User reaction to the pure HSV display is mixed; some find it compelling, whereas others find it difficult to parse the velocity field even given the continuous display of the HSV cone. For this reason, we augment the HSV display by blending it with a random texture map in which an intensity correlation has been induced upon flow lines. Methods for creating such an effect include the use of streak-lines, spot noise, and line integral convolution (*LIC*).<sup>10-12</sup> After considerable experimentation, we selected an implementation based upon the *LIC* algorithm of Cabral and Needham.<sup>10</sup>

From an abstract perspective (*LIC*) is straightforward. The input to the algorithm consists of a two dimensional random texture  $T_i(x, y)$  and a vector field  $\vec{v}(x, y)$ . The *LIC* procedure creates an output texture,  $T_o(x, y)$  in which each element,  $T_o(x_0, y_0) = T(\vec{s}_0)$ , is computed by the line integral

$$T_o(\vec{s}_0) = \int_{-\infty}^{\infty} k(\vec{s} - \vec{s}_0) T_i(\vec{\sigma}(\vec{s})) d\vec{s}$$

where  $k(\vec{s})$  is an averaging kernel such that  $0 \leq k(\vec{s})$  and  $\int_{-\infty}^{+\infty} k(\vec{s}) d\vec{s} = 1$ , and  $\sigma(\vec{s})$  is a curve that passes through  $\vec{s}_0$  and whose tangent at any point  $\vec{s}$  is given by  $\vec{v}(\vec{s})$ .<sup>11</sup> The averaging effect of the convolution produces a visible correlation among pixels lying along the streamlines associated with the vector field.

In application, it is necessary to replace the integral by a discrete summation. Generically, this process may be represented as

$$T_o(p_0) = \sum_{p \in N_{p_0}} T_i(p) h_{p_0}(p)$$

where  $p_0$  represents a pixel in the output texture,  $N_{p_0}$  is some neighborhood of  $p_0$ , and  $h$  is a weighting function such that  $\sum_{p \in N_{p_0}} h_{p_0}(p) = 1$ .

In Cabral and Leedom's algorithm, the output texture is viewed as a grid of square cells in which the center of each pixel corresponds to the center of a cell. The path over which the line integration takes place is a piecewise linear construction in which each linear segment crosses one of the square cells in the direction of the vector field at the center of that cell. Thus,  $N_{p_0}$  is the set of cells through which the piecewise linear path passes. For each cell,  $p$ , through which the path passes, the weight function  $h_{p_0}(p)$  is given by

$$h_{p_0}(p) = \int_{s_{entry}}^{s_{exit}} k(w) dw$$

where  $s_{entry}$  is the distance along the piecewise linear path from  $p_0$  to the point at which the path enters cell  $p$  and  $s_{exit}$  is the sum of  $s_{entry}$  and the linear segment that crosses cell  $p$  in the direction of the vector field. For the images shown here, a box filter,  $k(s) = \frac{1}{2L}$  for  $-L \leq s \leq L$  with  $L = 6$  was used. In blending the grayscale *LIC* texture with the HSV image a weight of 0.65 was used for the *LIC* texture and 0.35 for the HSV image. Values in the range [6, 20] produce images of comparable quantity, and thus  $L$  was chosen to minimize computational cost.

The *LIC* algorithm is also attractive because it is possible to animate the flow by simply replacing the box filter with a Hanning filter of the type  $k(s) = A(1 + \cos(\omega s + \phi))$ . Animation is achieved by incrementing  $\phi$  for each frame shown. An unexpected benefit of the blending operation was that the spatio-temporal artifacts associated with abrupt cutoff of  $k(s)$ <sup>10</sup> were reduced to such a degree that it was unnecessary to augment the Hanning filter with a Gaussian or Hanning window function.

## 8. CONCLUSIONS

We have presented a lattice-Boltzmann model augmented with a hue, saturation, value (HSV) color model and line integral convolution (*LIC*) for simulating and visualizing non-Newtonian fluid flows. We have provided a complete verification that the flow generated by the model is described by the Navier-Stokes equation. The visualization tool allows interactive changes to geometry of flow containers and barriers, and this, coupled with the near real-time model update, offers an effective tool for experimentation with the design of flow vessels, such as those found in the synthetic fiber and film industry.

## ACKNOWLEDGMENTS

This work was supported in part by the ERC Program of the U.S. National Science Foundation under award EEC-9731680 and the ITR Program of the National Science Foundation under award ACI-0113139.

## REFERENCES

1. E. Aharonov and D. Rothman, "Non-newtonian flow through porous media: A lattice-boltzmann method," *Geophysical Research Letters* **20**, pp. 679–682, April 1993.
2. L. Giraud, D. d'Humieres, and P. Lallemand, "A lattice boltzmann model for jeffreys viscoelastic fluid," *Europhysics Letters* **42**, pp. 625–630, June 1998.
3. I. Karlin, A. Ferrante, and C. Ottinger, "Perfect entropy functions of the lattice boltzmann method," *Europhysics Letters* **47**, pp. 182–188, July 1999.
4. U. Frisch, B. Hasslacher, and Y. Pomeau, "Lattice-gas automata for the navier-stokes equation," *Physical Review Letters* **56**, pp. 1505–1508, April 1986.
5. B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems*, Cambridge University Press, Cambridge, UK, 1998.
6. A. R. Smith, "Color gamut transform pairs," in *Proc. ACM SIGGRAPH*, pp. 12–19, 1978.
7. P. O'Neil, *Advanced Calculus Pure and Applied*, Macmillan, New York, NY, 1975.
8. F. Higuera, S. Succi, and R. Benzi, "Lattice gas dynamics with enhanced collisions," *Europhysics Letters* **9**, pp. 345–349, June 1989.
9. C. Henze, "Feature detection in linked derived spaces," in *Proc. IEEE Vis98*, pp. 87–94, (Raleigh, NC), October 1998.
10. B. Cabral and L. Leedom, "Imaging vector fields using line integral convolution," in *Proc. ACM SIGGRAPH 1993*, pp. pp. 263–270, August 1993.
11. W. de Leeuw and R. van Liere, "Comparing lic and spot noise," in *Proceedings of Visualization '98, IEEE Computer Society Press*, pp. 359–365, 1998.
12. D. Stalling and H. Hege, "Fast and resolution independent line integral convolution," in *Proc. ACM SIGGRAPH 1995*, pp. pp. 249–256, August 1995.