The Lattice-Boltzmann Method on Optimal Sampling Lattices

Usman R. Alim, Alireza Entezari, and Torsten Möller, *Member*, *IEEE*

Abstract—In this paper, we extend the single relaxation time Lattice-Boltzmann Method (LBM) to the 3D body-centered cubic (BCC) lattice. We show that the $D3bQ15$ lattice defined by a 15 neighborhood connectivity of the BCC lattice is not only capable of more accurately discretizing the velocity space of the continuous Boltzmann equation as compared to the $D3Q15$ Cartesian lattice, it also achieves a comparable spatial discretization with 30 percent less samples. We validate the accuracy of our proposed lattice by investigating its performance on the 3D lid-driven cavity flow problem and show that the $D3bQ15$ lattice offers significant cost savings while maintaining a comparable accuracy. We demonstrate the efficiency of our method and the impact on graphics and visualization techniques via the application of line-integral convolution on 2D slices as well as the extraction of streamlines of the 3D flow. We further study the benefits of our proposed lattice by applying it to the problem of simulating smoke and show that the $D3bQ15$ lattice yields more detail and turbulence at a reduced computational cost.

Index Terms—Visual simulation, animation, physically based modeling, BCC, volume modeling, vector field data, flow visualization, optimal regular sampling.

1 Introduction

In the context of signal processing and sampling theory, the goal is to sample a function densely enough (Nyquist), such that all frequencies in the spectrum of the function are properly captured by the discrete sampling. Assuming a band-limited function with an isotropic spectrum, the densest sphere packing of the spectra in the frequency domain introduces the best sampling lattice in the space domain [23]. Hexagonal packing of the 2D spectrum in the frequency domain leads to the dual hexagonal lattice being the best sampling pattern for sampling a bivariate function. Similarly, densely packing the 3D spectra in the frequency domain on the Face Centered Cubic (FCC) lattice introduces its dual, the *Body Centered Cubic* (BCC) lattice, as the best generic pattern for sampling trivariate functions. It is well known that the overall improvement of the BCC lattice over the the Cartesian lattice is 30 percent (either 30 percent less samples or 30 percent more information).

This advantage has been known for several decades now [23], [9], but has not yet found widespread acceptance in the graphics and visualization community. The reasons were twofold. On one hand, no practical reconstruction filters or other signal processing and analysis tools had been developed for the BCC lattice. Therefore, no devices or algorithms were built nor developed to acquire data directly on the BCC lattice. On the other hand, since there were no data given on the BCC lattice, there was no need to develop signal processing tools to process such lattices.

There has been an effort in the last several years to break this cycle. Entezari et al. have created proper reconstruction (interpolation, approximation) filters for the BCC lattice, which are comparable to their Cartesian counterparts in numerical accuracy [12], perceptual accuracy [21], and have been shown to be twice as fast as their Cartesian counterparts [13]. These insights make it almost possible to break the above-mentioned cycle. The one remaining roadblock for a more widespread acceptance of the BCC lattice in graphics and visualization has been the fact that there are no algorithms nor devices available that acquire sampled data directly on this lattice.

The main two sources of discretized volumetric data for volume graphics are medical acquisition devices and numerical simulations. In this paper, we focus on the numerical simulation of fluid flow, as expressed in the Navier-Stokes (NS) equations. The application of such algorithms cover a wide range of areas from visual effects [14], [15] to urban security [26].

The Lattice-Boltzmann method (LBM) is a promising alternative to traditional top-down techniques for solving the incompressible NS equations. It has gained attention in the visualization community in the past few years due to its easy implementation on graphics hardware. Since the purpose of this paper is to change the discretization lattice of the continuous Lattice-Boltzmann equation, we found it important to present an abridged derivation of these equations in Section 3. The novelty of this paper, however, is a careful consideration of the discretization of the velocities and its impact on the lattice structure (see Section 4) as well as their use in graphics and visualization.

The main contributions of this paper are the following:

- This is the first paper to acquire simulation data directly on a BCC lattice without a necessary resampling step. The advantage of employing this

- *U.R. Alim and T. Möller are with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, BC V5A 1S6, Canada. E-mail: {ualim, torsten}@cs.sfu.ca.*
- *A. Entezari is with the Department of Computing and Information Science and Engineering, University of Florida, E301 CSE Building, PO Box 116120, Gainesville, FL 32611. E-mail: entezari@cise.ufl.edu.*

Manuscript received 15 Sept. 2008; accepted 9 Dec. 2008; published online 31 Dec. 2008.
Recommended for acceptance by M. Chen, C. Hansen, and A. Pang.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCGSI-2008-09-0153.
Digital Object Identifier no. 10.1109/TVCG.2008.201.

pattern for simulation is not only on its improved accuracy of sampling, but also on its superior computational efficiency properties [13].

- Whereas in the traditional LBM, the spatial discretization lattice is almost always the Cartesian lattice, velocity discretization is chosen to include extra directions in addition to the axis-oriented directions. Our proposed lattice for spatial discretization, naturally provides more directions for discretization of velocity vectors. The increased symmetries in the proposed lattice complements, in the LBM, the sampling theoretical advantages such as the efficiency of reconstruction.
- We demonstrate the advantages of solving the LBM directly on a BCC lattice by improved efficiency of the simulations without incurring any loss in accuracy. Furthermore, stability is significantly enhanced in our proposed method. We provide the explanations for the improvements the BCC lattice brings to the LBM. We show the impact using line-integral convolutions on 2D slices of the simulation as well as 3D streamline computations and the simulation and rendering of smoke.

2 Related Research

The numerical solution of partial differential equations is a rich field with many challenges [22]. Modern sophisticated techniques often resort to multigrid methods and, in general, nonregular lattices. However, the simulation of natural phenomena (such as smoke, fire, or fluids) for visual effects in the movie or gaming industry is often done on a Cartesian lattice.

In the early days of Computer Graphics (CG), when the appearance of these phenomena was more important than physical accuracy, researchers used procedural methods [10]. Later, as computers became faster, top-down Computational Fluid Dynamics (CFD) techniques were used to generate images with high levels of visual realism. Foster et al. [15] provided a finite-difference solution to the NS equations and Fedkiw et al. [14] proposed the addition of a vorticity confinement force to address the problem of energy dissipation in coarse grids.

2.1 The Lattice-Boltzmann Method

Recently, bottom-up approaches have been introduced and consist of the Lattice Gas Cellular Automata (LGCA) and the LBM. The starting point of these approaches is a discrete mesoscopic model, which by construction, yields the Navier-Stokes equations of fluid flow. Both LGCA and the LBM have been used in the field of CG as an alternative to the top-down CFD-based techniques. Dobashi et al. [8] have used LGCA to produce realistic animations of clouds. Wei et al. have used the LBM to compute velocity fields for the purpose of fire simulation [32], create wind fields [34], and simulate gaseous phenomena [33].

A significant advantage that the LBM has over the top-down approaches is its simplicity. Computation of the LBM consists of inexpensive addition, subtraction, and multiplication operations. Furthermore, these operations are performed locally which can potentially improve computation speed by parallelization. Different types of boundaries can also be incorporated without the need to change the local computation rules.

2.2 Optimal Regular Lattices

All applications of the LBM use Cartesian type lattices for the spatial and velocity-space discretization of the Boltzmann equation. Such a discretization is not optimal in the sense of the isotropy of directions and the distribution of lattice nodes.

The book by Conway and Sloan has an exhaustive look at a number of different regular lattices and their properties [3]. A formal treatment of signal processing concepts for regular lattices, including the Cartesian and nonseparable lattices in 2D can be found in the book by Dudgeon and Mersereau [9]. After this early work, the 2D hexagonal lattice has not received much attention until Van De Ville et al. created a family of B-splines on the hexagonal lattice, also known as hex-splines [31].

The design of reconstruction kernels for the BCC lattice builds upon box splines [5], [12]. While this work had focused on the numerical accuracy of the reconstruction, it is important to evaluate these splines in terms of their perceptual behavior. In setting up a user study, where the users had to judge the quality of images reconstructed on a Cartesian pipeline versus a BCC pipeline, it has been established that the BCC lattice is comparable to a Cartesian lattice with only about 70 percent of the samples [21].

Despite the numerical accuracy and perceptual fidelity of these new box splines, they were prohibitively expensive in their evaluation [12]. Looking to make these splines practical for rendering applications, a surprising result was found. With proper piecewise polynomial evaluation of the spline, the rendering speed of a ray-caster, based on the BCC lattice using the quintic box spline, is twice the speed of an efficient implementation of a ray-caster on the Cartesian lattice based on cubic B-splines [13]. The intuition behind this result is the fact that the support of these (nonseparable) box splines is much more compact than separable, tensor-product splines.

3 The Lattice-Boltzmann Model

In contrast to the top-down CFD models that describe macroscopic fluid properties, the LBM is a bottom-up approach that describes the behavior of a fluid at a mesoscopic level. At this level, fluid flow is modeled by tracking the evolution of averaged distributions of microscopic particles. Such an approach avoids the complexities involved in dealing with fluid flow at a completely microscopic level and yet accurately predicts the macroscopic behavior of the flow. As compared to top-down models, the LBM has the advantage that the evolution rules that govern the distribution of particles are linear and local, thus making the method simpler and faster. For these reasons, the LBM is a well-suited candidate for solving fluid flow problems in computer graphics.

In this section, we will present a general description of the model that is not tied to any particular lattice structure. For a careful analysis and derivation of the LBM, we refer the reader to the excellent texts by Succi [29] and Wolf-Gladrow [35].

3.1 The Boltzmann Equation

As the name suggests, the LBM has its roots in the kinetic theory of gases where distribution functions such as the *Maxwell-Boltzmann* distribution [35] play a key role. For a domain $\Omega \subset \mathbb{R}^d$ $(d \in \{2,3\})$, a single species distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ is a function of the form $f : \Omega \times \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ and gives the continuous time-dependent probability of finding a fluid particle within an infinitesimal volume located at $\boldsymbol{x}$ and having velocity $\boldsymbol{v}$. The kinetic theory of gases deals with the time evolution of such distribution functions and relates macroscopic quantities such as fluid density, velocity, and pressure to the underlying microscopic particle distributions.

Under constant temperature conditions and when no external forces are acting on the fluid, the distribution function evolves according to the *Boltzmann Equation*

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f = Q(f, f), \tag{1}$$

where $\boldsymbol{v}$ is the velocity of a particle at the location $\boldsymbol{x}$ at time t. The two-particle collision term $Q(f, f)$ on the right-hand side of (1) models the way in which the distribution changes as a result of two particles colliding with each other. The term $\boldsymbol{v} \cdot \nabla f$ in (1) models the change in the distribution function as a result of the propagation of particles owing to their motion.

The collision term in the continuous Boltzmann (1), in general, has a complex integral form. In most applications of kinetic theory to fluid dynamics, the collision integral is approximated by simpler expressions. One such approximation is the BGK approximation proposed by Bhatnagar et al. and incorporated into the so-called *Lattice BGK* models by Qian et al. [25]. This approximation is motivated by the fact that the large amount of detail of the two-particle collision integral $Q(f, f)$ does not significantly influence the values of quantities at the macroscopic scale. Consequently, the two-particle term $Q(f, f)$ is replaced by a simpler term $J(f)$, given by

$$J(f) = \frac{1}{\tau}[f^{eq}(\boldsymbol{x}, \boldsymbol{v}, t) - f(\boldsymbol{x}, \boldsymbol{v}, t)]. \tag{2}$$

This operator models the effect of collisions as a relaxation of the distribution toward a Maxwellian equilibrium $f^{eq}(\boldsymbol{x}, \boldsymbol{v}, t)$. The parameter τ has dimensions of time and controls the frequency with which the distribution function relaxes to equilibrium. The BGK approximation is also sometimes referred to as the *single relaxation time* approximation and is quite popular in the LBM literature. Although there are other forms of the collision integral in use (see, e.g., [6] for a description of a *multiple relaxation time* collision model), our present work employs the BGK approximation as it recovers the NS equations in the appropriate macroscopic limits and is also suitable for use with benchmark problems.

3.2 LBGK Equation

The velocity space in the Boltzmann equation is continuous, i.e., a particle is allowed to move about freely with any velocity. A first step toward a discrete approximation of (1) is the restriction of the continuous velocity space to a finite set of velocities $\mathbf{V} := \{\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{n-1}\}$. This means that a particle located at $\boldsymbol{x}$ is now restricted to have one of the n velocities $\boldsymbol{c}_i$. The distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ now reduces to the function $f_i(\boldsymbol{x}, t)$ that describes distributions over a finite lattice of velocities. The Boltzmann equation under the BGK approximation becomes the *discrete Boltzmann equation* given by

$$\frac{\partial f_i}{\partial t} + \boldsymbol{c}_i \cdot \nabla f_i = \frac{1}{\tau}\left(f_i^{eq} - f_i\right). \tag{3}$$

This equation is still not suitable for computational use since it is continuous in space and time. In the LBM, (3) is discretized spatially and temporally such that the domain spacing $\Delta \boldsymbol{x}$ and the time step Δt are related by $\frac{\Delta \boldsymbol{x}}{\Delta t} = \boldsymbol{c}_i$. This discretization ensures that particles located at a discrete location (or node) $\boldsymbol{x}$ move within a time step Δt to a neighboring node $\boldsymbol{x} + \boldsymbol{c}_i \Delta t$ that is along the velocity vector $\boldsymbol{c}_i$. Taking the time step Δt to be unity, (3) becomes the *lattice BGK equation*, given by

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) - f_i(\boldsymbol{x}, t) = \frac{1}{\tau}\left(f_i^{eq}(\boldsymbol{x}, t) - f_i(\boldsymbol{x}, t)\right). \tag{4}$$

A simpler interpretation of this equation over a 2D or 3D discrete lattice is as follows:

1. *Collision.* At time t, particles at a discrete node $\mathbf{x}$ collide with each other, and in the process, change the distribution function to $f_i^*(\boldsymbol{x}, t)$, where

$$f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) + \frac{1}{\tau}(f_i^{eq}(\boldsymbol{x}, t) - f_i(\boldsymbol{x}, t)). \tag{5}$$

2. *Propagation.* Within one time step, the postcollisional distribution values $f_i^*(\boldsymbol{x}, t)$ propagate to their neighboring nodes along the lattice velocities $\boldsymbol{c}_i$. More formally,

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) = f_i^*(\boldsymbol{x}, t). \tag{6}$$

This step does not alter the distributions as they are propagating. Therefore, by construction, mass is conserved.

It is worth pointing out that the collision and propagation rules are simple and linear. The collision step at a particular lattice node is entirely local and does not need information from any other nodes. During the propagation step, distribution values propagate to the neighbors along the lattice velocities. This step therefore updates the distribution function at at most n nodes. Due to the simplicity and locality of these rules, the LBM is readily amenable to parallelization.

3.3 Macroscopic Quantities

The variables of interest in most fluid flow applications are the macroscopic fluid density and velocity. In the LBM, these quantities are obtained by ensemble averages of the underlying particle distribution function $f_i(\boldsymbol{x}, t)$. The density of the fluid at a discrete node $\boldsymbol{x}$ is given by the total number of particles that reside at the node while the fluid velocity is given by the first velocity moment of the distribution function

$$\rho(\boldsymbol{x}, t) = \sum_i f_i(\boldsymbol{x}, t), \tag{7}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho}\sum_i f_i(\boldsymbol{x}, t)\boldsymbol{c}_i. \tag{8}$$

3.4 Lattice Symmetry and Equilibrium Distribution

In order to recover the incompressible NS equations in the macroscopic limit, the discrete set of velocities used in the LBM must respect certain symmetry constraints as outlined in [35], [17]. The first constraint is that the velocity set itself is symmetric, i.e.,

$$\mathbf{V} = -\mathbf{V}. \tag{9}$$

The second set of constraints is related to the isotropy of a lattice. Let W_i be a weight associated with a lattice velocity

$\boldsymbol{c}_i$. Moreover, let us assume that the weights are the same for lattice velocities having the same speed and that they satisfy the relation

$$\sum_i W_i = 1. \tag{10}$$

Isotropy requirements impose the constraints that the generalized lattice tensors of rank two and four must be isotropic, i.e.,

$$\sum_i W_i \boldsymbol{c}_{i\alpha}\boldsymbol{c}_{i\beta} = c_s^2 \delta_{\alpha\beta}, \tag{11}$$

$$\sum_i W_i \boldsymbol{c}_{i\alpha}\boldsymbol{c}_{i\beta}\boldsymbol{c}_{i\gamma}\boldsymbol{c}_{i\delta} = c_s^4\left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right), \tag{12}$$

where the summation is taken over all lattice velocities. The Greek subscripts refer to the different Cartesian components of a lattice velocity $\boldsymbol{c}_i$ and $\delta_{\alpha\beta}$ is the Kronecker delta symbol. The parameter c_s is the sound speed of a fluid in equilibrium and is a constant for a given lattice.

The equilibrium particle distribution $f_i^{eq}(\boldsymbol{x}, t)$ depends on the local fluid density and velocity, and has the implicit form $f_i^{eq}(\rho(\boldsymbol{x}, t), \boldsymbol{u}(\boldsymbol{x}, t))$. It is computed by taking the velocity moments of the Maxwell-Boltzmann distribution and equating it with the respective moments of the equilibrium distribution $f_i^{eq}(\rho, \boldsymbol{u})$. For velocity lattices that satisfy the above constraints, this procedure results in a closed-form expression for the equilibrium particle distribution given by

$$f_i^{eq}(\rho, \boldsymbol{u}) = \rho W_i \left[1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2c_s^2}\right]. \tag{13}$$

By construction, the equilibrium distribution conserves mass, i.e.,

$$\sum_i f_i^{eq} = \rho. \tag{14}$$

3.5 Recovery of Navier-Stokes Equations

No special treatment is needed to obtain solutions to the macroscopic equations of fluid flow. It has been shown [25], [35] that the fluid density and velocity obtained through the LBM ((7) and (8)) solve the incompressible Navier-Stokes equations

$$\nabla \cdot \boldsymbol{u} = \boldsymbol{0}, \tag{15}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{u}\cdot\nabla u + \nu\nabla^2\boldsymbol{u} - \nabla p, \tag{16}$$

where ν is the viscosity of the fluid and p is the scalar pressure. The viscosity of the fluid is related to the sound speed and the collision frequency through

$$\nu = \frac{1}{2}c_s^2(2\tau - 1). \tag{17}$$

Note that the constant mass density of the fluid is taken to be unity and does not appear in the NS equations. When the fluid speed is sufficiently small as compared to the sound speed of the lattice ($\frac{|\boldsymbol{u}|}{c_s} \ll 1$), the collision and propagation rules of the LBM ensure that deviations of the fluid density from its initial value are negligible. The LBM therefore solves the Navier-Stokes equations in the *incompressible limit*.

3.6 Initial and Boundary Conditions

Since the variables of interest are the macroscopic quantities, the LBM needs to be able to handle initial and boundary conditions that are prescribed in terms of the fluid density and velocity. The general approach for handling initial conditions is to set the initial distribution function $f_i^0(\boldsymbol{x})$ equal to the Maxwellian distribution that satisfies the initial fluid density $\rho_0(\boldsymbol{x})$ and the initial fluid velocity $\boldsymbol{u}_0(\boldsymbol{x})$:

$$f_i^0(\boldsymbol{x}) = f_i^{eq}(\rho_0(\boldsymbol{x}), \boldsymbol{u}_0(\boldsymbol{x})), \quad \boldsymbol{x} \in \Omega. \tag{18}$$

Starting from an initial particle distribution over the lattice, the system evolves according to the collision and propagation rules defined by (5) and (6). While the collision rule is well defined for all lattice sites, the propagation rule is not well defined for lattice nodes that have one or more neighbors that either lie outside the domain Ω or are obstructed by a solid obstacle inside the fluid. In such cases, the propagation rule needs to be redefined so that the prescribed macroscopic boundary conditions are satisfied.

A very appealing property of the LBM is the fact that different types of boundary conditions can easily be incorporated into the model simply by modifying the distributions locally. Fluid inlets and outlets as well as slip and no-slip boundary conditions can all be modeled in terms of the distribution values alone. However, care must be taken in doing so since errors due to inaccurate boundary treatments propagate to other parts of the domain, thus affecting the solution everywhere.

4 Sampling and Discretization

In the LBM literature, the notation $DdQn$ is usually used to indicate the choice of a discrete set of velocities at every lattice point. The variable d implies the spatial dimension where the simulation occurs and n is the number of discrete velocities. The spatial and velocity discretizations are related to each other in that, once a lattice has been chosen, the velocity discretization is implicitly given by choosing a particular neighborhood around a lattice point. The arrangement of the velocities in the neighborhood must respect the symmetry and isotropy requirements outlined in Section 3.4 so that the incompressible NS equations can be recovered in the macroscopic limit.

As an example, in 2D, the $D2Q7$ and the $D2Q9$ configurations are commonly used (Fig. 1). The nodes in $D2Q7$ are arranged on a hexagonal lattice and the velocity discretization is obtained by connecting each point to itself and six of its nearest neighbors. In contrast, the nodes in $D2Q9$ are arranged on a Cartesian lattice and the velocity discretization is obtained by connecting a node to itself, four nearest neighbors that are a unit distance away in the axis-aligned directions, and four second-nearest neighbors that are at a distance of $\sqrt{2}$ in the diagonal directions. The hexagonal lattice is of primary importance in signal processing where it is known to achieve the highest packing density of Fourier-transform replicas of an isotropically band-limited bivariate function [9]. As can be seen in Fig. 1a, the $D2Q7$ configuration leads to a more isotropic velocity discretization where the Voronoi cells of the six neighbors on the hexagonal lattice share a face with the Voronoi cell of the center. On the other hand, we have a mix of face and vertex connectivity in the Cartesian case (Fig. 1b). Two key questions are in order at this

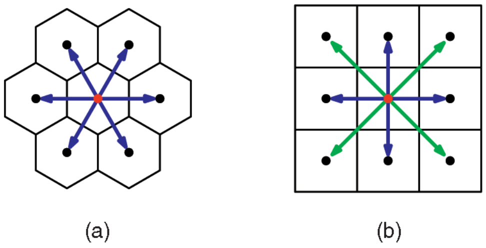

Fig. 1. The 2D LBM lattices. (a) The $D2Q7$ Lattice. (b) The $D2Q9$ Lattice.

point: what ramifications does the optimal sampling property have for the LBM and, if there are any advantages, do they hold in 3D as well?

4.1 Cartesian and BCC Lattices

In order to answer the above questions, let us proceed by first introducing the 3D lattices. The Cartesian lattice is the usual choice for discretizing the Boltzmann equation in 3D. The velocity space is then discretized by choosing an appropriate sublattice of the Cartesian lattice (Fig. 2a). With respect to a lattice site, we refer to the nearest neighbors as first-order neighbors, the second-nearest neighbors as second-order neighbors, and so on.

The $D3Q15$ lattice consists of a central node, 6 first-order neighbors given by the permutations of $(\pm1, 0, 0)$ and 8 third-order neighbors given by the permutations of $(\pm1, \pm1, \pm1)$. The $D3Q19$ lattice consists of the central node and the 6 first-order neighbors as well as 12 second-order neighbors given by the permutations of $(\pm1, \pm1, 0)$. The $D3Q27$ lattice consists of the central node and all the first, second, and third-order neighbors. Besides these three lattices, another lattice that is commonly used is $D3Q13$ formed by the central node and the 12 second-order neighbors. The velocity discretization of $D3Q13$ leads to the structure of an FCC lattice. However, it is usually used in conjunction with a Cartesian lattice where simulations are carried out on two decoupled FCC lattices that form a Cartesian lattice [7].

The BCC lattice is the optimal sampling lattice in 3D. It is a sublattice of the Cartesian lattice $\mathbb{Z}^3$ formed by retaining all points whose coordinates have the same parity, i.e either all the coordinates are odd or all of them are even. The resulting lattice is four times less dense giving the Voronoi cell (a Truncated Octahedron), a volume of 4 [11]. A lattice point and 14 of its neighbors that share a face of the Truncated Octahedron with it, give us a 15-point discretization which we refer to as $D3bQ15$ (Fig. 2b). Out of these 14 neighbors, 8 are first-order given by offsets obtained through the combinations of $(\pm h, \pm h, \pm h)$ and 6 are second-order given by offsets obtained through the permutations of $(\pm 2h, 0, 0)$, where h is an arbitrary scaling parameter. In contrast to $D3Q15$ which has a mix of face and vertex neighbors, $D3bQ15$ only has face neighbors which makes it a more isotropic structure. It is easily verified that the $D3bQ15$ lattice indeed satisfies the isotropy requirements needed to recover the NS equations. We give the derivation of the weights and the speed of sound c_s in Appendix A. Also summarized in Appendix A are some other important properties of the various 3D lattices that we shall make use of.

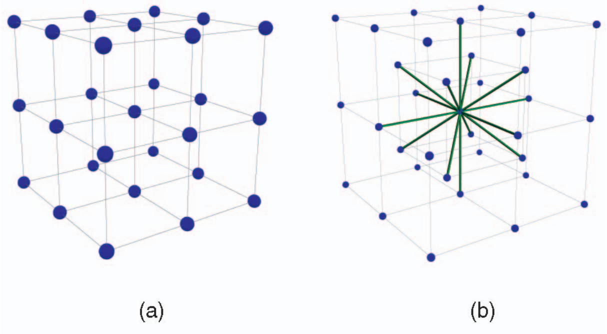

Fig. 2. The 3D simulation lattices. (a) The Cartesian Lattice, a 27-point view. (b) The BCC Lattice showing a 14-neighbor connectivity.

The lattice scaling parameter h for the $D3bQ15$ lattice can be chosen in one of two ways. Either the lattice can be scaled so that the speed of sound is the same as that of the Cartesian lattices (i.e., $1/\sqrt{3}$), or it can be scaled such that the Voronoi cell has a unit volume resulting in a sampling density that is the same as the Cartesian lattice $\mathbb{Z}^3$. Henceforth, we refer to the former case as $D3bQ15$ and the latter case as $D3bQ15^*$. Since the BCC lattice is four times less dense as compared to $\mathbb{Z}^3$, $h = 1/\sqrt[3]{4}$ for $D3bQ15^*$. Remarkably, for $D3bQ15$, the scaling parameter turns out to be $h = 1/\sqrt{2}$, which is precisely the scaling factor that allows a BCC lattice to capture the frequency information of an isotropically band-limited trivariate function with 30 percent less samples as compared to its Cartesian counterpart [11]. A similar result also holds for the 2D case where the $D2Q7$ lattice scaled to have the same speed of sound as $D2Q9$ yields a 14 percent saving in samples.

4.2 Spatial and Velocity Discretization

The discrete propagation step (6) provides a second-order accurate spatial discretization of the continuous Boltzmann equation in a particular direction. Errors introduced as a result of this step can be lowered by reducing the distance between lattice points. Table 1 gives the average distance between a lattice point and its neighbors for various 3D lattices. $D3bQ15$, being a first-second-order configuration, is better than $D3Q15$ which is only a first-third-order configuration. Moreover, it achieves a lower average distance with 30 percent less samples. $D3bQ15^*$, due to its higher sampling density, provides an even better discretization, which is comparable to that of $D3Q19$ which is also a first-second-order configuration.

It should be noted that the accuracy of the solution at a point not only depends on the spatial discretization error but also on the velocity discretization error. Configurations with more neighbors will generally tend to have lower velocity discretization errors. Also, when the number of velocities is the same, we would expect that configurations

TABLE 1
Average Distance between a Lattice Point and Its Neighbors

Lattice Type	Average Distance
$D3Q15$	1.418
$D3bQ15$	1.306
$D3bQ15^*$	1.163
$D3Q19$	1.276
$D3Q27$	1.416

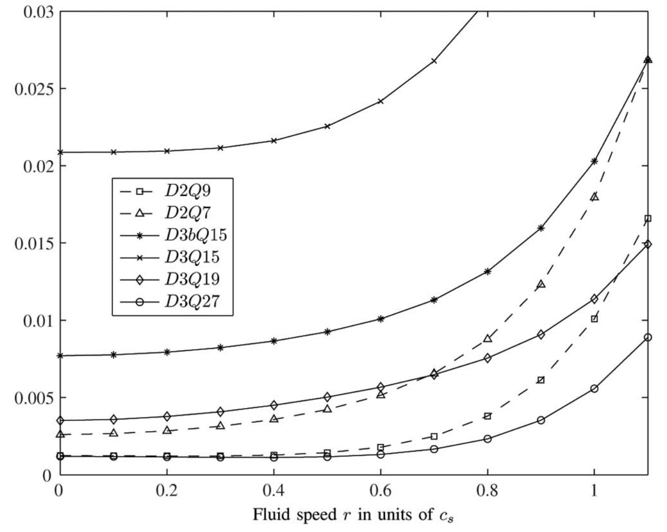

Fig. 3. Total approximation error $\triangle(r)$ for various lattices. $D2Q7$ refers to the 2D scaled hexagonal lattice that has the same speed of sound as $D2Q9$. For all lattices shown, $c_s = 1/\sqrt{3}$. For each lattice velocity, a Monte-Carlo integration method with 100,000 samples was used to evaluate (20). In 2D, the integrals are computed over the boundary of a circle of radius r.

that are more isotropic should have lower velocity discretization errors. Thus, $D3bQ15$ and $D3bQ15^*$ should be better than $D3Q15$. To formalize this notion, let us turn back to the collision step of the LBM (5) which relies on the discrete equilibrium distribution function (13). The accuracy of the collision step therefore depends on the error introduced by approximating the equilibrium distribution function on a particular lattice. Equation (13) can be seen as a second-order Taylor-series expansion (see [35] for details) of the 3D Boltzmann distribution given by

$$f_i^M(\boldsymbol{u}) = \left(\frac{1}{2\pi c_s^2}\right)^{\frac{3}{2}} \exp\left[-\frac{(\boldsymbol{u}-\boldsymbol{c}_i)^2}{2c_s^2}\right]. \tag{19}$$

The constant mass density is assumed to be unity and does not appear in the above equation. Let us define the approximation error in a particular lattice direction for a fluid moving with speed r as

$$\Delta_i(r) := \frac{1}{4\pi r^2}\int_{\Omega_r}\left(f_i^M(\boldsymbol{u}) - f_i^{eq}(1,\boldsymbol{u})\right)^2 dS, \tag{20}$$

where Ω_r refers to the surface of the sphere of radius r. Using this, we define the total approximation error as

$$\triangle(r) := \sum_i \Delta_i(r). \tag{21}$$

Fig. 3 shows plots of the function $\triangle(r)$ for various lattices and various fluid speeds. A general trend observed for all lattices is that the error is low when the fluid speed is small, which confirms the fact that the LBM recovers the NS equations in the low-Mach-number regime ($r \ll c_s$). On the Cartesian lattice, $D3Q15$ is by far the worst. With larger neighborhoods, $D3Q19$ and $D3Q27$ provide much better approximations for the equilibrium distribution function. On the BCC lattice, with the same number of velocities, $D3bQ15$ is significantly better than its Cartesian counterpart $D3Q15$ and only slightly worse than $D3Q19$. $D3Q27$ achieves the lowest error in 3D which suggests that a corresponding 27-neighbor configuration on the BCC lattice should perform similarly.

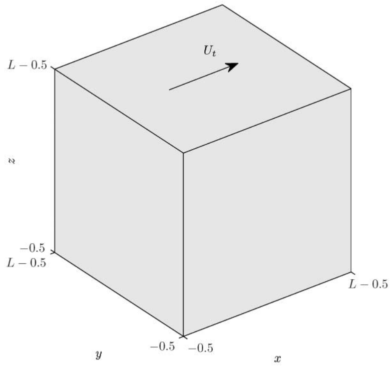

Fig. 4. Schematic of the lid-driven cavity.

Thus, we observe that the sampling optimality of a lattice has immediate consequences for the LBM. Not only does $D3bQ15$ provide a much better velocity-space discretization as compared to $D3Q15$, it also provides a better spatial discretization with 30 percent less samples.

5 3D Cavity Flow

We use the 3D lid-driven cavity flow problem to compare the accuracy of the LBM on Cartesian and BCC lattices. This problem has been extensively studied and is commonly used as a benchmark to compare the accuracy of incompressible NS solvers. See, e.g., [1], [4], [19] for results obtained through top-down methods and [20] for results obtained through the LBM. In our present work, we focus on the $D3Q15$ and $D3Q19$ Cartesian configurations, and compare them with the $D3bQ15$ and $D3bQ15^*$ BCC configurations.

An incompressible viscous fluid is enclosed in a cubic cavity of side $L (L \in \mathbb{Z})$ and is driven into motion by a lid moving at constant speed. Since the problem is symmetric in space, it does not matter which face of the cube is chosen as the driving lid. In our simulations, the cavity is arranged as shown in Fig. 4 and the top face is chosen as the lid which moves in the positive x-direction with a speed of $U_t = 0.1$. The other faces of the cavity are treated as solid walls that satisfy a no-slip boundary condition. The Reynolds number of the flow is denoted as Re and is given by $Re = U_t L/\nu$, where ν is the kinematic viscosity of the fluid.

For the Cartesian configurations, the distribution of nodes is straightforward. A cavity of side length L has L^3 nodes that lie completely inside the cavity with the boundary being 0.5 units away in each direction, as shown in Fig. 4. For the purpose of computation, the nodes can easily be stored as a 3D array. A BCC grid can also be efficiently stored in memory as a 3D array [30], where xy slices having an odd index are shifted by $(h, h, 0)$. Each xy slice itself is a Cartesian grid with a spacing of $2h$ units in the x and y-directions respectively, the distance between successive slices in the z-direction being h. With this arrangement scheme, the resolution of the BCC grid is $N_x \times N_y \times N_z$, where

$$N_x = N_y = \left\lfloor \frac{L-h}{2h} + 1 \right\rfloor, \quad N_z = \left\lfloor \frac{L}{h} + 1 \right\rfloor, \tag{22}$$

and the nodes are arranged so that they are centered within the cavity. $D3bQ15$ and $D3bQ15^*$ configurations are obtained by choosing h appropriately, as discussed earlier.

The viscosity of the fluid for a given Reynolds number is computed as $\nu = U_t L/Re$ and the relaxation parameter τ is then chosen according to (17). Initially, all nodes except the ones that lie in the topmost xy slice are initialized to an equilibrium distribution corresponding to a fluid with density $\rho = 1$ and zero velocity (13). The top nodes are initialized to an equilibrium distribution of $\rho = 1$ and $\boldsymbol{u} = (U_t, 0, 0)$.

One time step of the simulation then proceeds by applying the collision and propagation rules ((5) and (6)) to each of the lattice nodes. At the end of a time step, each node is updated to its modified density and velocity computed through the distributions that have propagated to the nodes ((7) and (8)).

In order to ensure a constant driving speed of the lid, an inlet boundary condition is imposed at all the top nodes. This is achieved by resetting the distribution function at all the top nodes, at the end of each time step, to the initial equilibrium distribution. On the other walls of the cavity, we use a simple bounce-back scheme in order to satisfy the no-slip boundary condition. We note that our choice of boundary conditions is similar to that used by Hou et al. [16] for the 2D lid-driven-cavity problem but differs from those used by Mei et al. [20] for the corresponding 3D problem.

In order to compare the performance of numerical methods on this benchmark problem, the usual practice is to compare the steady-state velocity profiles along the axial directions in the geometric center of the cavity. For this reason, we run our simulations until a steady velocity field is obtained. We consider the velocity field to be steady if

$$\max \|\boldsymbol{u}(\boldsymbol{x}, t+1) - \boldsymbol{u}(\boldsymbol{x}, t)\| < \epsilon, \tag{23}$$

where the convergence threshold ϵ is taken to be 10^{-5}. We denote the x-component of the steady-state velocity field as $u(\boldsymbol{x})$, the y-component as $v(\boldsymbol{x})$, and the z-component as $w(\boldsymbol{x})$.

Visualization. We also compare the quality of the simulation results by visualizing the steady-state vector fields using Line Integral Convolution (LIC) [2], [27] in 2D and streamlines in 3D. We use a second-order Runge-Kutta integrator to perform streamline integration. During streamline integration, the vector fields need to be interpolated at nongrid points. For the Cartesian simulation, we use a tricubic B-spline interpolation scheme, and for the BCC simulation, we use a quintic box spline interpolation scheme [12], [13], thus ensuring that the approximation order is the same.

5.1 Results and Discussion

We carried out cavity flow simulations at different Reynolds numbers using different lattice sizes. We only consider the case of $Re \leq 1{,}000$ since, in this regime, cavity flow becomes steady. In order to compare the velocity profiles, all our results are normalized so that the cavity lies within the unit cube $[0,1]^3$, and the velocity is scaled by a factor of $1/U_t$. This normalization also allows us to compare our results with the benchmark data of Albensoeder and Kuhlmann [1] and the results of Mei et al. [20].

In order to gain some insight into the performance of the different lattices, we first compare them at $Re = 1{,}000$ and $L = 96$. Accuracy at a high Reynolds number is a good indicator of the overall performance of a numerical scheme as accuracy generally deteriorates with increasing Re. Fig. 5 shows three axial profiles for the different lattices tested. The results for $D3Q15$ do not appear on the plots since the corresponding simulation did not converge even though a high lattice resolution was used. Excellent agreement is obtained between $D3Q19$ and the BCC configurations $D3bQ15$ and $D3bQ15^*$ despite the fact that $D3bQ15$ is 30 percent coarser. However, we observe that our results do not agree with the benchmark results and differ with respect to the location of the velocity extrema. We consistently observe such differences across all test cases and attribute the disagreement to the fact that our boundary treatment for the driving lid is different from Mei et al. [20], who have used a modified bounce-back scheme to satisfy a Dirichlet boundary condition on the lid wall. We believe that a Dirichlet boundary condition on the lid wall is a more accurate description of the problem and conforms well to the boundary conditions used by top-down methods.

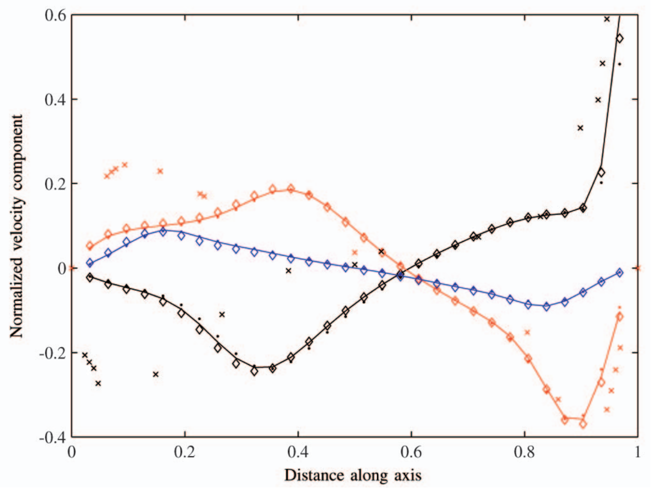

Fig. 5. Axial profiles at $Re = 1{,}000$ and $L = 96$ through the center of the cavity. The lattices are denoted as $D3Q19$ (solid lines), $D3bQ15$ (◇), and $D3bQ15^*$ (●). Colors indicate different profiles; black: $u(\frac{1}{2},\frac{1}{2},z)$; blue: $v(\frac{1}{2},y,\frac{1}{2})$; and red: $w(x,\frac{1}{2},\frac{1}{2})$. The symbols (×) are the results of Albensoeder and Kuhlmann [1]. Note that the benchmark results for the blue profile are not provided in [1]. The resolutions are $D3Q19$ $(96 \times 96 \times 96)$, $D3bQ15$ $(68 \times 68 \times 136)$, and $D3bQ15^*$ $(76 \times 76 \times 153)$ calculated using (22).

The effect of changing the Reynolds number is illustrated in Fig. 6. As Re increases, the flow becomes more turbulent and the profiles show more high-frequency features. Again, simulation results for BCC agree very well with those of Cartesian. We notice that there is a comparatively greater difference near the top boundary which we believe is due to the fact that the geometry of the Cartesian lattice is better able to represent the straight boundaries of the cavity.

Fig. 7 shows the effect that lattice resolution has on accuracy at $Re = 400$. At this Reynolds number, the simulations for both $D3bQ15$ and $D3Q15$ did not converge for a coarse resolution lattice. However, $D3bQ15^*$ which has the same computational burden as $D3Q15$ not only converges, but also yields a velocity profile that is closer to the one obtained for $D3Q19$ at a higher resolution.

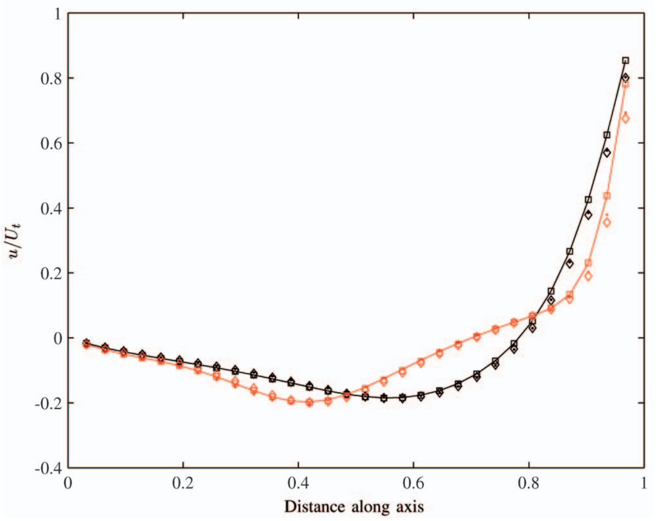

Fig. 6. Velocity profile $u(\frac{1}{2},\frac{1}{2},z)$ at $Re = 100$ (black) and $Re = 400$ (red) with $L = 64$. The symbols are as given in Fig. 5, $D3Q15$ is denoted by (□).

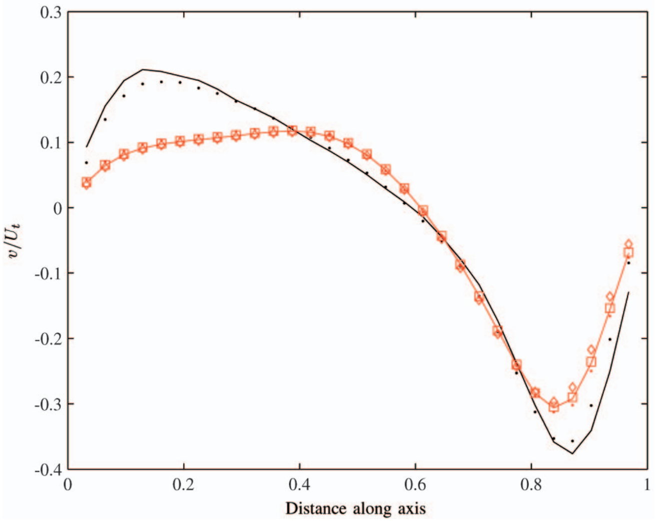

Fig. 7. Velocity profile $v(\frac{1}{2},y,\frac{1}{2})$ for $L = 32$ (black) and $L = 64$ (red) at $Re = 400$. The symbols are as given in Figs. 5 and 6.

Thus, we see that $D3Q15$ is the least stable. With the same number of velocities, $D3bQ15$ and $D3bQ15^*$, due to their better spatial and velocity discretization, perform much better than $D3Q15$. Our results suggest that $D3bQ15^*$, by virtue of its higher sampling density, is comparable to $D3Q19$ both in terms of accuracy and stability. On the other hand, $D3bQ15$, having a lower sampling density, is not as stable as $D3bQ15^*$. However, in terms of accuracy, it performs favorably when the parameters are well within the stability regime of the problem, thus allowing one to speed up simulations by 30 percent.

Finally, Fig. 8 shows a qualitative comparison of the flow field at $L = 96$ and $Re = 1,000$. At this Reynolds number, two vortices emerge as illustrated by the LIC slices (Figs. 8a, 8b, and 8c). These slices are color mapped to show the strength of the flow field. Due to the differences in the lid boundary between CC and BCC, we observe that $D3Q19$ exhibits a stronger flow near the top as compared to the BCC lattices, $D3bQ15$ and $D3bQ15^*$. There is also a slight disagreement in the location of the primary and secondary vortices. The nature of the flow around the primary vortex is shown in Figs. 8d, 8e, and 8f. Again, the images are virtually indistinguishable, but we do note that $D3bQ15^*$ is closer to $D3Q19$ than $D3bQ15$. Since it is more efficient to perform interpolations on a BCC lattice [13], we observe that all our BCC visualizations incur a reduced computational cost as compared to CC.

6 Smoke Simulation

In order to demonstrate the advantages of the LBM on BCC lattices, we apply it to the problem of simulating smoke visually. Wei et al. [33] have used the LBM for this purpose. However, they did not incorporate the effects of an external body force into the LBM which limited the levels of turbulence that they were able to achieve. For this reason, we follow the recipe of Fedkiw et al. [14] with the modification that we use the LBM as a NS solver.

We model smoke as a nonreactive substance that is suspended in a viscous, incompressible fluid and gets advected by the flow. We denote the time-dependent density of smoke as $P(\boldsymbol{x},t)$ and assume that it obeys the advection equation

$$\frac{\partial P}{\partial t} = -\boldsymbol{u} \cdot \nabla P, \tag{24}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ is the time-dependent velocity of the fluid.

Furthermore, smoke exerts a buoyancy force on the fluid in the $+z$-direction. We model this force as

$$\boldsymbol{f}_{\mathrm{buoy}}(\boldsymbol{x},t) = \alpha P(\boldsymbol{x},t)\hat{\boldsymbol{k}}, \tag{25}$$

where α is a tunable parameter that controls the strength of the force.

The effects of an external force can be incorporated in the LBM by modifying the collision step. The propagation step remains unaffected. We use the body-force model of Junk et al. [18] according to which the collision step (5) changes to

$$f_i^*(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + \frac{1}{\tau}\left(f_i^{eq}(\boldsymbol{x},t) - f_i(\boldsymbol{x},t)\right) + 3W_i\boldsymbol{c}_i \cdot \boldsymbol{G}(\boldsymbol{x},t), \tag{26}$$

where W_i are the lattice weights and $\boldsymbol{G}(\boldsymbol{x},t)$ is a time-dependent external force.

Smoke simulation proceeds by initially setting the fluid mass density to ρ_0 and giving it a constant upward velocity of magnitude U_0. These values are then used to initialize the packet distributions in the LBM. Before the simulation begins, certain grid points are designated as smoke sources. The density of smoke is initialized to zero at all points that are not smoke sources and to a value P_0 at the smoke sources. Smoke is reinjected into the sources at the end of each time step throughout the simulation. Like the cavity flow problem, the viscosity of the fluid is determined from the Reynolds number according to $\nu = U_0L/Re$, where L is the vertical height of the simulation domain. Similarly, a no-slip boundary condition is imposed on all the boundary nodes through the use of the bounce-back rule, in effect simulating smoke trapped within a solid box.

After initialization, the simulation proceeds iteratively for a user-specified number of time steps. At the end of

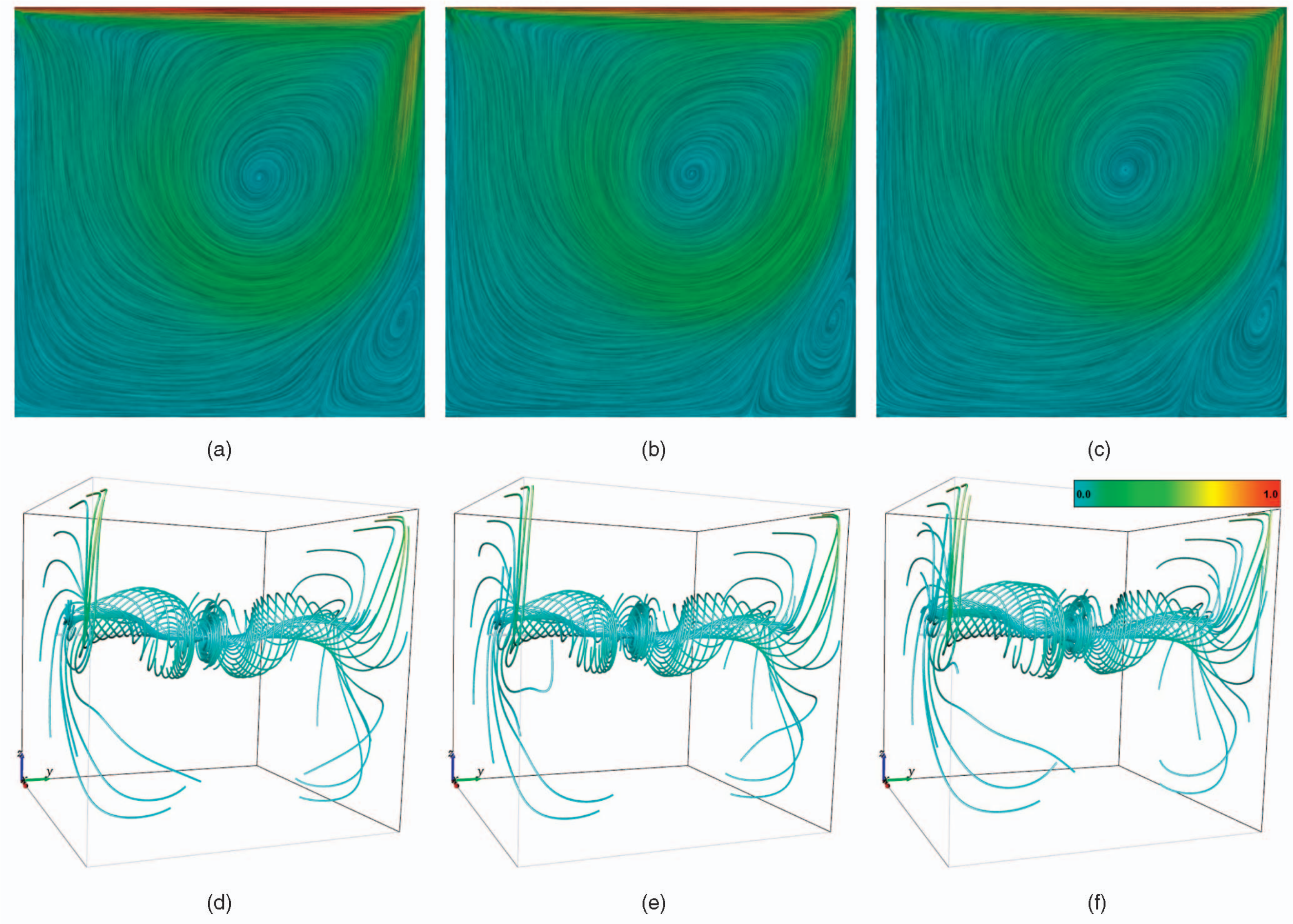

Fig. 8. LIC and streamline images at $Re = 1,000$ and $L = 96$. (a) and (d) $D3Q19$, (b) and (e) $D3bQ15$, (c) and (f) $D3bQ15^*$. The LIC images (a), (b), and (c) illustrate the flow on the xz plane in the center of the cavity. Streamline images (d), (e), and (f) show the flow around the primary vortex. As illustrated, a line source is used to seed the streamlines. Refer to the accompanying supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2008.201, for animations showing streamlines emanating from the line source as it moves through the cavity. The LIC computation time averaged over 10 runs is (a) 44.7 s, (b) 37.7 s, and (c) 39.0 s. The average streamline computation time per line is (d) 3.41 ms, (e) 2.92 ms, and (f) 2.89 ms.

each time step, we use the macroscopic velocity of the fluid in a second-order Runge-Kutta-based semi-Lagrangian integration scheme [28] to advect the smoke density. In order to avoid the problem of overshooting, we use trilinear interpolation on CC and linear box spline interpolation on BCC [12]. We then use the macroscopic fluid velocity and the advected smoke density to calculate a vertical buoyancy force (25) as well as a vorticity confinement force given by

$$\boldsymbol{f}_{\text{vort}}(\boldsymbol{x}, t) = \epsilon(\boldsymbol{N} \times \boldsymbol{\omega}), \tag{27}$$

where ω is the vorticity (curl) of the velocity field and N is the normalized gradient of the magnitude of vorticity. The parameter ϵ controls the strength of the vorticity confinement force. We refer the reader to [14] for details on how this force addresses the problem of decaying turbulence in smoke simulations. The external force acting at each grid point is simply the sum of the buoyancy and vorticity confinement forces, i.e.,

$$\boldsymbol{G}(\boldsymbol{x}, t) = \boldsymbol{f}_{\text{buoy}}(\boldsymbol{x}, t) + \boldsymbol{f}_{\text{vort}}(\boldsymbol{x}, t), \tag{28}$$

and is used in the subsequent collision step to drive the flow. Since the BCC grid has axis-aligned neighbors in the three axial directions, we use a central-difference scheme on both CC and BCC to approximate the various partial derivatives needed for the computation of the vorticity confinement force. We note that this is not the optimal choice for BCC and other more accurate strategies are possible.

Lastly, we visualize the smoke density using a single-scattering ray-casting procedure [24]. The extinction coefficient σ_t is directly computed from the density of the smoke through $\sigma_t = C_t P$, where C_t is the extinction cross section and P is the smoke density. In order to achieve better quality, we make use of higher order (tricubic B-spline on CC and quintic box spline on BCC [13]) schemes to interpolate the smoke density.

6.1 Results and Discussion

Simulating smoke using the LBM is a challenging problem in that the range of parameters over which the model is stable is quite limited as compared to traditional top-down solvers. The nature of the flow is also affected by the shape of the simulation domain as the solid walls influence the flow globally. Furthermore, since the LBM is a low-Mach-number model, the magnitude of the external force must be kept low in order to ensure stability.

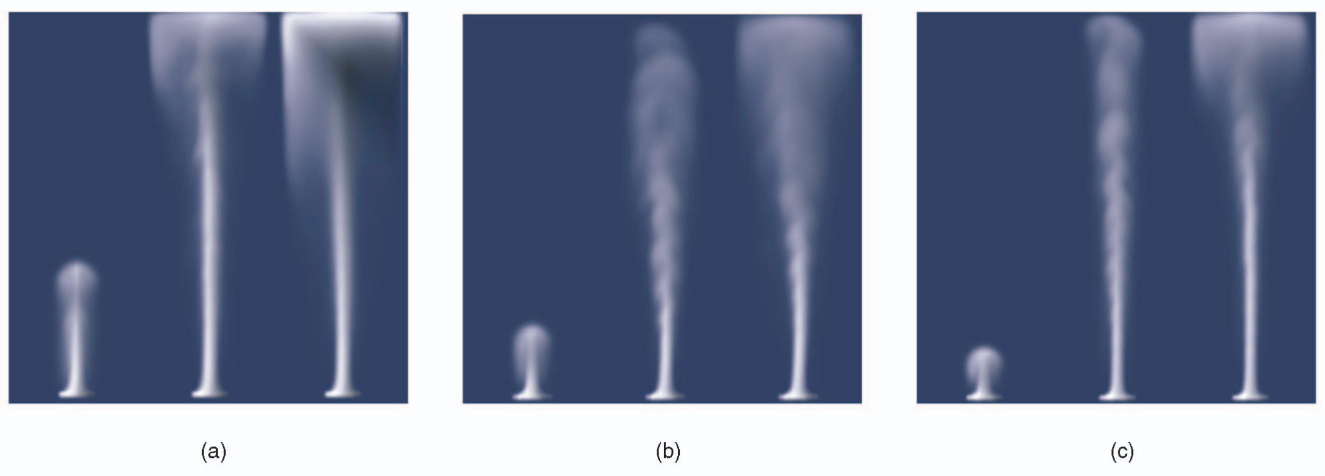

Fig. 9. Three frames of an animation showing smoke rising from a single source at $Re = 300$. (a) $D3Q19$, (b) $D3bQ15$, and (c) $D3bQ15^*$. The parameters used are $\rho_0 = 2.7$, $U_0 = 0.1$, $\alpha = 1.0 \times 10^{-4}$, and $\epsilon = 0.1$. All images were rendered with a vertical resolution of 512 pixels. Grid resolution, average simulation time per iteration, and average rendering time per frame are (a) $40 \times 20 \times 128, 0.390$ s, 54.3 s, (b) $28 \times 14 \times 180, 0.245$ s, 34.4 s, and (c) $31 \times 16 \times 202, 0.356$ s, 35.3 s.

We conducted all our simulations and renderings on a machine with two AMD dual-core Opteron 280 processors with 8 GB of memory. Fig. 9 shows how smoke emerges from a single source placed at the bottom of a vertical column at $Re = 300$. Grid resolution as well as simulation and rendering timings are indicated. Also indicated are the initial fluid density and velocity, and the parameters that control the magnitude of the buoyancy and vorticity confinement forces. At this Reynolds number, both $D3bQ15$ and $D3bQ15^*$ as well as $D3Q19$ are stable whereas $D3Q15$ is unstable. Since the flow changes quite gradually between each iteration, we used the density field after every 20 iterations for the purpose of producing the next animation frame.

Placing the source close to the walls of the column gives rise to turbulence, thereby causing the smoke to swirl as it rises. The job of the vorticity confinement force is to ensure that this swirling motion does not decay over time. We notice that the BCC configurations (Figs. 9b and 9c) preserve the vorticity of the flow much better than $D3Q19$ (Fig. 9a) which exhibits lesser vorticity and a faster vertical flow. This decaying turbulence also comes at the expense of greater simulation and rendering times. On the other hand, the BCC configurations not only exhibit greater vorticity, they do so at a reduced computational cost.

Increasing the Reynolds number to 400 while keeping the other parameters constant results in decreased stability, $D3bQ15^*$ being the only stable configuration. Interestingly, we observe that increasing the grid resolution while holding the other parameters constant also results in reduced stability. Fig. 10 shows simulation results for $D3bQ15$ using a grid that has twice the resolution along each dimension. The corresponding simulation for $D3bQ15^*$ remains stable but becomes unstable for both the Cartesian configurations $D3Q15$ and $D3Q19$. We attribute this instability to the fact that a higher grid resolution yields a flow with a greater level of turbulence which, in turn, results in a stronger vorticity confinement force being applied during the collision step. Notice how in comparison to Fig. 9b, Fig. 10a shows more discernible small-scale features, specially near the top boundary. Fig. 10b illustrates how a more turbulent appearance can be achieved by increasing the Reynolds number. Notice how vorticity is preserved throughout the course of the simulation. Refer to the supplemental material for the accompanying animations, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2008-201.

7 Conclusion

In this paper, we extended the Lattice Boltzmann Method to the BCC lattice configurations $D3bQ15$ and $D3bQ15^*$, and compared them to the Cartesian configurations $D3Q15$ and $D3Q19$ using the 3D lid-driven cavity problem. We further investigated the advantage of our proposed configurations by applying them to the problem of visually simulating smoke.

We have shown that the sampling optimality of the BCC lattice has important ramifications for the LBM. $D3bQ15$ with a 15-neighbor connectivity that is scaled to have the same speed of sound as the Cartesian configurations yields comparable results at a 30 percent savings in samples. It is also more stable as compared to $D3Q15$. Stability is further improved on $D3bQ15^*$, which, due to its higher sampling density, reduces the spatial discretization error. These benefits are clearly demonstrated in our results. Smoke simulations on a BCC lattice yield higher degrees of turbulence at a reduced computational cost as compared to their Cartesian cousins.

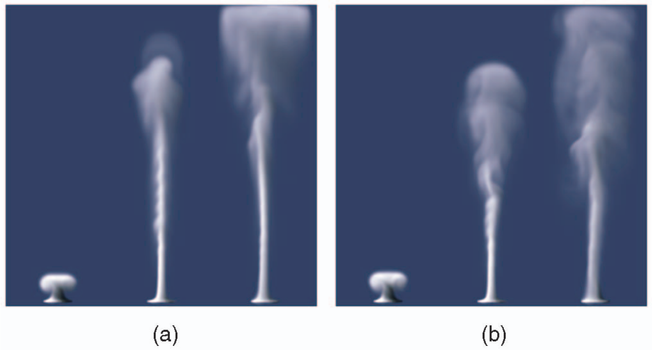

Fig. 10. Three frames of simulations conducted on $D3bQ15$ using a higher grid resolution of $56 \times 28 \times 361$. (a) $Rc = 300$ and (b) $Re = 400$.

The impact of the isotropy of the neighborhood on BCC leads us to believe that a 27-neighborhood connectivity should further improve the accuracy and stability of the model, thus allowing us to simulate more turbulent flows. In future, we plan to investigate this notion further, as well as address the problem of finding a set of neutral boundary conditions that do not favor the Cartesian lattice.

While previous research has shown that the BCC lattice is a much preferred lattice for the representation, rendering, and reconstruction of continuous data, this is the first paper that acquires such data directly on a BCC lattice. We further show that the BCC lattice is the preferred lattice for solving the Navier-Stokes equation using the Lattice Boltzmann Method, not only because of its accuracy and stability, but also because of the fact that flow data acquired on the BCC lattice can be visualized much more efficiently.

Appendix A

Lattice Properties

Derivation of Lattice Weights for BCC

The $D3bQ15$ lattice consists of three different speeds; hence, there are three different weights: w_1, w_2, and w_3. w_1 corresponds to the zero velocity $(0,0,0)$, w_2 corresponds to the 8 first-order velocities $(\pm h, \pm h, \pm h)$, and w_3 corresponds to the 6 second-order velocities given by the permutations of $(\pm 2h, 0, 0)$. Let us denote the speed of sound as c_s. Conservation of mass gives us the first equation:

$$w_1 + 8w_2 + 6w_3 = 1.$$

The lattice tensor of rank 2 (11) is isotropic and yields the equation

$$8h^2(w_2 + w_3) = c_s^2.$$

Lattice tensor of rank 4 (12) is isotropic and yields the two equations:

$$\begin{aligned} 8h^4(w_2 + 4w_3) &= 3c_s^4, \\ 8h^4 w_2 &= c_s^4. \end{aligned}$$

The above four equations can be solved for w_1, w_2, w_3, and c_s. The solution is

$$w_1 = \frac{7}{18}, \quad w_2 = \frac{1}{18}, \quad w_3 = \frac{1}{36}, \quad c_s = \sqrt{\frac{2}{3}}h.$$

Miscellaneous Properties

Some of the important properties of the lattices that we have used are summarized below. L is a particular lattice and V refers to the volume of its Voronoi cell. The weights are given in ascending order according to the order of the velocities.

L	V	c_s	**Weights**
$D3Q15$	1	$\frac{1}{\sqrt{3}}$	$w_1 = \frac{2}{9}$, $w_2 = \frac{1}{9}$, $w_3 = \frac{1}{72}$
$D3bQ15$	$\sqrt{2}$	$\frac{1}{\sqrt{3}}$	see above
$D3bQ15^*$	1	$\frac{1}{\sqrt[6]{2}\sqrt{3}}$	see above
$D3Q19$	1	$\frac{1}{\sqrt{3}}$	$w_1 = \frac{1}{3}$, $w_2 = \frac{1}{18}$, $w_3 = \frac{1}{36}$
$D3Q27$	1	$\frac{1}{\sqrt{3}}$	$w_1 = \frac{8}{27}$, $w_2 = \frac{2}{27}$, $w_3 = \frac{1}{54}$, $w_4 = \frac{1}{216}$

Acknowledgments

This work has been made possible in part by the support of the Canadian Foundation of Innovation (CFI) and the Natural Science and Engineering Research Council of Canada (NSERC). The authors would like to thank the anonymous reviewers who helped improve this paper through their feedback and comments.

References

[1] S. Albensoeder and H.C. Kuhlmann, "Accurate Three-Dimensional Lid-Driven Cavity Flow," *J. Computational Physics,* vol. 206, no. 2, pp. 536-558, 2005.

[2] B. Cabral and L.C. Leedom, "Imaging Vector Fields using Line Integral Convolution," *Proc. ACM SIGGRAPH '93,* pp. 263-270, 1993.

[3] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups,* third ed. Springer, 1999.

[4] A. Cortes and J. Miller, "Numerical Experiments with the Lid Driven Cavity Flow Problem," *Computers and Fluids,* vol. 23, no. 8, pp. 1005-1027, 1994.

[5] C. de Boor, K. Höllig, and S. Riemenschneider, *Box Splines.* Springer Verlag, 1993.

[6] P. Dellar, "Incompressible Limits of Lattice Boltzmann Equations Using Multiple Relaxation Times," *J. Computational Physics,* vol. 190, no. 2, pp. 351-370, 2003.

[7] D. d'Humières, M. Bouzidi, and P. Lallemand, "Thirteen-Velocity Three-Dimensional Lattice Boltzmann Model," *Physical Rev. E,* vol. 63, no. 6, p. 66702, 2001.

[8] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita, "A Simple, Efficient Method for Realistic Animation of Clouds," *Proc. 27th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 19-28, 2000.

[9] D.E. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing,* first ed. Prentice-Hall, Inc., 1984.

[10] D. Ebert and R. Parent, "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-Buffer Techniques," *Proc. 17th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 357-366, 1990.

[11] A. Entezari, "Optimal Sampling Lattices and Trivariate Box Splines," PhD dissertation, Simon Fraser Univ., Vancouver, Canada, July 2007.

[12] A. Entezari, R. Dyer, and T. Möller, "Linear and Cubic Box Splines for the Body Centered Cubic Lattice," *Proc. IEEE Conf. Visualization,* pp. 11-18, Oct. 2004.

[13] A. Entezari, D. Van De Ville, and T. Möller, "Practical Box Splines for Reconstruction on the Body Centered Cubic Lattice," *IEEE Tran. Visualization and Computer Graphics,* vol. 14, no. 2, pp. 313-328, Mar./Apr. 2008.

[14] R. Fedkiw, J. Stam, and H.W. Jensen, "Visual Simulation of Smoke," *Proc. ACM SIGGRAPH '01,* pp. 15-22, 2001.

[15] N. Foster and D. Metaxas, "Modeling the Motion of a Hot, Turbulent Gas," *Proc. 24th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 181-188, 1997.

[16] S. Hou, Q. Zou, S. Chen, G. Doolen, and A. Cogley, "Simulation of Cavity Flow by the Lattice Boltzmann Method," *J. Computational Physics,* vol. 118, no. 2, pp. 329-347, 1995.

[17] M. Junk, A. Klar, and L. Luo, "Asymptotic Analysis of the Lattice Boltzmann Equation," *J. Computational Physics,* vol. 210, no. 2, pp. 676-704, 2005.

[18] M. Junk and Z. Yang, "One-point Boundary Condition for the Lattice Boltzmann Method," *Physical Rev. E,* vol. 72, no. 6, p. 66701, 2005.

[19] H. Ku, R. Hirsh, and T. Taylor, "A Pseudospectral Method for Solution of the Three-Dimensional Incompressible Navier-Stokes Equations," *J. Computational Physics,* vol. 70, no. 2, pp. 439-462, 1987.

[20] R. Mei, D. Yu, and L. Luo, "Lattice Boltzmann Method for 3D Flows with Curved Boundary," *J. Computational Physics,* vol. 161, no. 2, pp. 680-699, 2000.

[21] T. Meng, B. Smith, A. Entezari, A.E. Kirkpatrick, D. Weiskopf, L. Kalantari, and T. Möller, "On Visual Quality of Optimal 3D Sampling and Reconstruction," *Proc. Graphics Interface Conf. '07,* pp. 265-272, May 2007.

[22] K. Morton and D. Mayers, *Numerical Solution of Partial Differential Equations.* Cambridge Univ. Press, 1994.

[23] D.P. Petersen and D. Middleton, “Sampling and Reconstruction of Wave-Number-Limited Functions in + *N*-Dimensional Euclidean Spaces,” *Information and Control,* vol. 5, no. 4, pp. 279-323, Dec. 1962.

[24] M. Pharr and G. Humphreys, *Physically Based Rendering: From Theory to Implementation.* Morgan Kaufmann, 2004.

[25] Y. Qian, D. D’Humières, and P. Lallemand, “Lattice BGK Models for Navier-Stokes Equation,” *Europhysics Letters,* vol. 17, 479, 1992.

[26] F. Qiu, Y. Zhao, Z. Fan, X. Wei, H. Lorenz, J. Wang, S. Yoakum-Stover, A. Kaufman, and K. Mueller, “Dispersion Simulation and Visualization for Urban Security,” *Proc. IEEE Conf. Visualization ’04,* pp. 553-560, 2004.

[27] D. Stalling and H.-C. Hege, “Fast and Resolution Independent Line Integral Convolution,” *Proc. ACM SIGGRAPH ’95,* pp. 249-256, 1995.

[28] J. Stam, “Stable Fluids,” *Proc. ACM SIGGRAPH ’99,* pp. 121-128, 1999.

[29] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford Univ. Press, 2001.

[30] T. Theußl, T. Möller, and E. Gröller, “Optimal Regular Volume Sampling,” *Proc. IEEE Visualization Conf. ’01,* pp. 91-98, Oct. 2001.

[31] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle, “Hex-Splines: A Novel Spline Family for Hexagonal Lattices,” *IEEE Trans. Image Processing,* vol. 13, no. 6, pp. 758-772, June 2004.

[32] X. Wei, W. Li, K. Mueller, and A. Kaufman, “Simulating Fire with Texture Splats,” *Proc. IEEE Visualization Conf. ’02,* pp. 227-235, 2002.

[33] X. Wei, W. Li, K. Mueller, and A. Kaufman, “The Lattice-Boltzmann Method for Simulating Gaseous Phenomena,” *IEEE Trans. Visualization and Computer Graphics,* vol. 10, no. 2, pp. 164-176, Mar./Apr. 2004.

[34] X. Wei, Y. Zhao, Z. Fan, W. Li, S. Yoakum-Stover, and A. Kaufman, “Blowing in the Wind,” *Proc. ACM SIGGRAPH/ Eurographics Symp. Computer Animation,* pp. 75-85, 2003.

[35] D.A. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction.* Springer-Verlag Telos, 2000.

Usman R. Alim received the BS and BA degrees in physics and mathematics from the University of Rochester and the MS degree in computer science from Rochester Institute of Technology. He is currently working toward the PhD degree at the School of Computing Science, Simon Fraser University. His research interests span the fields of computer graphics and visualization, and include global illumination, physically-based modeling, and the applications of optimal sampling lattices.

Alireza Entezari received the PhD degree from the School of Computing Science, Simon Fraser University, Canada, in 2007. He is an assistant professor at the Computer and Information Science and Engineering Department, University of Florida. His research interests include reconstruction of trivariate functions, optimal sampling lattices, multivariate spline approximation and interpolation, and their applications in numerical computing, visualization, and computer graphics.

Torsten Möller received the Vordiplom (BSc) in mathematical computer science from Humboldt University of Berlin, Germany, and the PhD degree in computer and information science from Ohio State University in 1999. He is an associate professor at the School of Computing Science, Simon Fraser University. He is the director of Vivarium, the codirector of the Graphics, Usability and Visualization Lab (GrUVi), and serves on the Board of Advisors for the Centre for Scientific Computing at Simon Fraser University. His research interests include the fields of visualization and computer graphics, especially the mathematical foundations thereof. He is the appointed Vice Chair for Publications of the IEEE Visualization and Graphics Technical Committee (VGTC). He has served on a number of program committees (including the Eurographics and the IEEE Visualization conferences) and has been papers cochair for EuroVis, Graphics Interface, and the Workshop on Volume Graphics as well as the Visualization track of the 2007 International Symposium on Visual Computing. He has also co-organized the 2004 Workshop on Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration at the Banff International Research Station. He is currently serving on the steering committee of the Symposium on Volume Graphics. He is an associate editor for the *IEEE Transactions on Visualization and Computer Graphics* (*TVCG*) as well as the *Computer Graphics Forum*. He is a member of the IEEE, the ACM, the Eurographics, and the Canadian Information Processing Society (CIPS).