

2008

Methods for stabilizing high Reynolds number Lattice Boltzmann simulations

Brian Robert Keating
College of William & Mary - Arts & Sciences

Follow this and additional works at: <https://scholarworks.wm.edu/etd>



Part of the [Plasma and Beam Physics Commons](#)

Recommended Citation

Keating, Brian Robert, "Methods for stabilizing high Reynolds number Lattice Boltzmann simulations" (2008). *Dissertations, Theses, and Masters Projects*. Paper 1539623525.
<https://dx.doi.org/doi:10.21220/s2-m0sb-5082>

This Dissertation is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Dissertations, Theses, and Masters Projects by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.

Methods for Stabilizing High Reynolds Number Lattice Boltzmann
Simulations

Brian Robert Keating

Hyattsville, Maryland

Master of Science, College of William & Mary, 2005
Bachelor of Science, Saint Mary's College of Maryland, 2003

A dissertation presented to the Graduate Faculty
of the College of William and Mary in Candidacy for the Degree of
Doctor of Philosophy

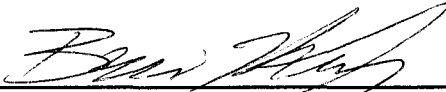
Department of Physics

The College of William and Mary
January, 2008

APPROVAL PAGE

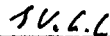
This Dissertation is submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy



Brian Robert Keating

Approved by the Committee, August, 2007



Committee Chair

Professor George Vahala, Physics
The College of William & Mary



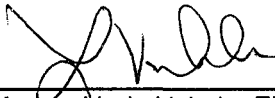
Assistant Professor Joshua Erlich, Physics
The College of William & Mary



Professor Kenneth Petzinger, Physics
The College of William & Mary



Professor Eugene Tracy, Physics
The College of William & Mary



Associate Professor Linda Vahala, Electrical Engineering
Old Dominion University

ABSTRACT PAGE

The Lattice Boltzmann Method (LBM) is a simple and highly efficient method for computing nearly incompressible fluid flow. However, it is well known to suffer from numerical instabilities for low values of the transport coefficients. This dissertation examines a number of methods for increasing the stability of the LBM over a wide range of parameters. First, we consider a simple transformation that renders the standard LB equation implicit. It is found that the stability is largely unchanged. Next, we consider a stabilization method based on introducing a Lyapunov function which is essentially a discrete-time H-function. The uniqueness of an H-function that appears in the literature is proven, and the method is extended to stabilize some of the more popular LB models. We also introduce a new method for implementing boundary conditions in the LBM. The hydrodynamic fields are imposed in a transformed moment space, whereas the non-hydrodynamic fields are shifted over from neighboring nodes. By minimizing population gradients, this method exhibits superior numerical stability over other widely employed schemes when tested on the widely-used benchmark of incompressible flow over a backwards-facing step.

For mom

Notation

Greek indices are used for Cartesian vector components, so for example, $\partial_\alpha \partial \frac{\partial}{\partial x_\alpha}$.

Latin subscripts (mostly i and j) index the lattice velocities. Summation is implied over repeated Greeks, but not over Latins. \vec{c} is used for microscopic (or molecular) velocities and \vec{u} denotes the macroscopic (fluid) velocity. Unless otherwise noted, all sums go from $1 \dots q$, that is, over all lattice velocities. Integrals over x are over the entire domain and integrals over \vec{c} are likewise over all velocities.

Contents

<i>Dedication</i>	<i>i</i>
<i>Notation</i>	<i>ii</i>
<i>List of Figures</i>	<i>v</i>
<i>List of Tables</i>	<i>vi</i>
Chapter 1 - Introduction	1
Chapter 2 - The Navier-Stokes Equations and Turbulence	3
2.1 <i>Conservation Equations</i>	3
2.2 <i>Fluid Equations</i>	5
2.3 <i>The Navier-Stokes Equations</i>	6
2.4 <i>Scalings for the Navier-Stokes Equations</i>	9
2.5 <i>Phenomenology of Turbulence</i>	12
2.6 <i>Kolmogorov Theory</i>	13
Chapter 3 - Numerical Solutions of Partial Differential Equations	17
3.1 <i>Parallel Computing</i>	17
3.2 <i>Numerical Stability</i>	20
3.3 <i>A Juxtaposition of Methods of Discretization</i>	21
Finite Differences (FD).....	21
Spectral Methods	22
Finite Volume Method.....	23
Finite Element Method (FEM).....	24
Chapter 4 - The Lattice Boltzmann Method	26
4.1 <i>Continuous Kinetic Theory</i>	26
4.2 <i>Lattice Boltzmann</i>	29
4.3 <i>The Chapman-Enskog Procedure</i>	35
4.4 <i>Lattice Boltzmann for the Navier-Stokes Equations</i>	37

Chapter 5 - An Implicit Lattice Boltzmann Scheme.....	43
Chapter 6 - Entropic Lattice Boltzmann	48
6.1 <i>Sub-grid Modeling</i>	48
6.2 <i>Reynolds Averaged Navier-Stokes</i>	49
6.3 <i>Large Eddy Simulations</i>	51
6.4 <i>Entropic Lattice Boltzmann</i>	53
The Newton-Raphson Method	56
6.5 <i>The Form of the H-function</i>	58
Zürich School	58
The Boston School	63
6.6 <i>Unified Entropic Lattice Boltzmann [31]</i>	67
6.7 <i>Simulations of 3-D Navier-Stokes Turbulence</i>	69
Chapter 7 - Moment Space Boundary Conditions.....	73
7.1 <i>Previous Boundary Condition Treatments</i>	75
Bounce-back	75
Zuo & He	76
Chen, Martinez and Mei	77
Diffusive Boundary Conditions	78
7.2 <i>Moment Space Boundary Conditions</i>	79
7.3 <i>Treatment of the Inlet/Outlet</i>	82
7.4 <i>Numerical Validation</i>	86
Chapter 8 - Conclusion	92
<i>Appendix</i>	95
<i>References</i>	96

List of Figures

Figure 1: <i>The components of the stress tensor</i>	7
Figure 2: <i>Kolmogorov energy spectrum for isotropic turbulence</i>	15
Figure 3: <i>The parallel efficiency of an LU linear solver</i>	19
Figure 4: <i>An example of spurious oscillations of a numerical solution in the vicinity of a sharp gradient</i>	21
Figure 5: <i>An approximate solution and basis functions for the linear finite element method</i>	25
Figure 6: <i>The relaxation of the distribution function</i>	28
Figure 7: <i>The lattice vectors of the D3Q19 model</i>	30
Figure 8: <i>The LB streaming process</i>	32
Figure 9: <i>A pictorial representation of the basic LB algorithm</i>	33
Figure 10: <i>The scaling of the computing power of a LB code with the number of processors used</i>	34
Figure 11: <i>The collision implicit algorithm</i>	45
Figure 12: <i>The initial conditions for the two-dimensional Taylor vortex</i>	46
Figure 13: <i>An arbitrary field before and after the application of an LES filter</i>	52
Figure 14: <i>The collision process in ELB for an arbitrary collision operator</i>	55
Figure 15: <i>A single iteration of the Newton-Raphson method</i>	57
Figure 16: <i>ELB-generated surfaces of constant vorticity magnitude at times $t = 0, 500, 1000$ and 1500 for $v \times 1.7 \times 10^{-4}$</i>	70
Figure 17: <i>The evolution of the kinetic energy and enstrophy for the 15- and 27-bit models using both entropic and non-entropic methods at higher viscosity</i>	71
Figure 18: <i>The maximum value of the x-component of vorticity as a function of time for the 15- and 27-bit models using both entropic and non-entropic methods at higher viscosity</i>	71

Figure 19: <i>The evolution of the kinetic energy and enstrophy for the 15- and 27-bit models using both entropic and non-entropic methods at lower viscosity</i>	72
Figure 20: <i>A generic 9-bit bottom boundary node</i>	74
Figure 21: <i>The populations at a boundary node before and after implementing no-slip bounce-back boundary conditions</i>	75
Figure 22: <i>The geometry for testing the inlet treatment</i>	84
Figure 23: <i>The evolution of the density/pressure at the point M with different inlet treatments</i>	84
Figure 24: <i>The evolution of the density/pressure at the point M with different outlet treatments</i>	86
Figure 25: <i>The geometry of the backwards-facing step</i>	87
Figure 26: <i>Snapshots of the velocity field for the backwards-facing step</i>	89
Figure 27: <i>The variation of reattachment length as a function of Re using various numerical methods</i>	90

List of Tables

TABLE I. <i>The lattice velocities and speeds of common LB models</i>	31
TABLE II. <i>The weights for the polynomial equilibrium populations</i>	39
TABLE III. <i>The weights that appear in the H-function for selected LB models</i>	62
TABLE IV. <i>The stability thresholds for backstep simulations using several boundary condition schemes</i>	89

Chapter 1

Introduction

Most flows are turbulent. In fluid flow problems, smooth, laminar flow is the exception to the turbulent rule. Turbulence is everywhere in nature, from the solar flares in the sun's atmosphere, to planet-wide motions of weather systems to cream being stirred into coffee. The irregular, stochastic quality of such flows has defied attempts to develop a unified theory of turbulence.

It is astonishing to realize that the vast array of complex fluid phenomena that we experience every day are all governed by quite simple sets of deterministic equations. Further, these equations can be derived only on the basis of conservation of mass and momentum (and possibly energy). The complicated phenomenology of turbulence is a result of nonlinear terms that appear in the equations of motion. It is this nonlinearity that makes fluid flow a mathematically challenging problem.

Only the simplest fluid flows can be solved analytically. When the geometry of the problem is sufficiently complicated, as is the case in most systems of practical interest, one must resort to numerical methods. The field of computational fluid dynamics (CFD) is vast; many methods for solving partial differential equations (PDE's) numerically have existed for decades and are briefly reviewed in Section 3.3.

In the last fifteen years or so, a new way to solve fluid equations numerically has been developed. In this approach, called the Lattice Boltzmann method (LBM), a

simplified kinetic equation is drastically discretized in velocity space in such a way that the correct fluid equations are obtained in the macroscopic limit. There are several advantages of working at the kinetic (or mesoscopic) level, including ease of implementation on parallel computers. Most importantly, in the LBM, the nonlinear derivative term in the fluid equations is replaced by simple advection at a constant velocity. By working in a higher-dimensional space, it is hoped that complicated particle trajectories will appear simpler.

A crucial stumbling block in the adoption of LB methods for realistic flows is numerical instability. Turbulent flows are characterized by a small viscosity. Like many other CFD methods, LB displays troubling numerical defects when the viscosity is made too small. This is due to the fact that a small viscosity implies that there are identifiable structures on a very large range of scales – too many to be directly captured by a simulation. This dissertation will present several different methods to extend the usefulness of the LBM to a wider range of parameters.

Chapter 2

The Navier-Stokes Equations and Turbulence

2.1 Conservation Equations

The mathematical laws of nature are usually cast in the form of partial differential equations. Such equations are formal embodiments of certain established general principles of physics. In particular, implicit in many physical laws is the temporal conservation of certain quantities – mass, momentum, energy, charge, etc. As such, they can be written in so-called *conservation form*

$$\frac{\partial A}{\partial t} + \nabla \cdot \vec{B} = 0, \quad (2.1)$$

where A is conserved quantity (or a vector of conserved quantities) and $\vec{B} = \vec{B}(A)$ is its flux.

The reason for this terminology can be demonstrated by considering an arbitrary control volume Ω whose boundary we denote by $\partial\Omega$. We integrate Eq. (2.1) over Ω

$$\int_{\Omega} \frac{\partial A}{\partial t} + \int_{\partial\Omega} \vec{B} \cdot \vec{n} = 0, \quad (2.2)$$

apply the divergence theorem and move the time derivative outside of the integral to obtain

$$\frac{d}{dt} \int_{\Omega} A \, d\Omega = \int_{\partial\Omega} \vec{B} \cdot \hat{n} \, dS, \quad (2.3)$$

where \hat{n} is the unit outward normal. Here we see that the time rate of change of the quantity A inside Ω is equal to the flux of A through its boundary. A can be transported in and out of a given volume, but cannot be created or destroyed.

The flux of a quantity can sometimes be given by the value of the quantity multiplied by a fluid velocity

$$\vec{B} = A \vec{u} \quad (2.4)$$

$$\frac{\partial A}{\partial t} + \nabla \cdot (A \vec{u}) = 0. \quad (2.5)$$

Such an equation expresses the perfect conservation of a continuum quantity in time as it is advected by a velocity field \vec{u} .

Often we find that the quantity A is not perfectly conserved, but is subject to a small amount of dissipation or dispersion. This can be expressed by adding a higher derivative to the RHS of Eq. (2.1)

$$\frac{\partial A}{\partial t} + \nabla \cdot \vec{B} - \nabla \cdot (\kappa \nabla^2 A), \quad (2.6)$$

where $\nabla \cdot \vec{B}$ represents dissipation (usually via heat) and $\nabla \cdot (\kappa \nabla^2 A)$ produces dispersion.

The coefficient κ is an example of a *transport coefficient*.

Examples of natural laws that can be expressed in the form given by Eq. (2.6) include the Navier-Stokes equations, shallow water equations, Korteweg-deVries

* The integral form of fluid equations is often considered more fundamental than the differential form since it allows for discontinuous solutions, i.e., shocks.

equation, Burger's equation, the magnetohydrodynamic (MHD) equations and many more. They describe the motion/evolution of continuum, fluid-like quantities and will be the focus of this dissertation.

2.2 Fluid Equations

As indicated in the previous section, fluid equations are based on the principles of conservation of mass and momentum (and energy, in the case of thermal flows). When the system's parameters are such that the molecular mean free path, l , is small compared to the system length scale, L ,

$$\frac{l}{L} \ll 1, \quad (2.7)$$

where Kn is called the Knudsen number, we can treat the system as a continuum. We can treat macroscopic variables as field quantities, which are differentiable and are defined throughout the domain.

Fluid equations are intentionally formulated to be insensitive to the underlying microscopic dynamics. Thus, for example, the Navier-Stokes equations which described general fluid flow have existed in their present form for over 150 years, and were unaffected by, for instance, the discovery of the atom, quantum mechanics, etc. Indeed, the basic form of fluid equations cannot change, since we know that under ordinary circumstances mass, momentum and energy are conserved.

Mesoscopic methods, including Lattice Boltzmann and its predecessor Lattice Gas Automata (LGA) [1], exploit the universality of fluid equations by devising

highly artificial microscopic dynamics in order to simplify computations. Provided that one respects certain microscopic conservation laws, one can recover the correct continuum equations in the long-wavelength, long-time limit, despite the fact that the simulated microscopic dynamics may not accurately reflect the true molecular dynamics.

2.3 The Navier-Stokes Equations

The Navier-Stokes equations describe a very broad class of neutral fluid flow problems which lack phase transitions. They can be derived from standard kinetic theory, but we will base our approach on (macroscopic) conservation laws. Many other systems of fluid equations are derived by starting with the Navier-Stokes equations and making various simplifying assumptions.

We will consider the conservation of mass and momentum of an infinitesimally small fluid element. Taking the generic conservation equation Eq. (2.1) as a starting point, we express the conservation of the mass density ρ of a fluid element by

$$\frac{\partial \kappa}{\partial t} + \partial_i (\kappa \bar{u}_i) = 0, \quad (2.8)$$

and the conservation of momentum, $\rho \bar{u}$, is expressed as

$$\frac{\partial \kappa \bar{u}_i}{\partial t} + \partial_j (\kappa \bar{u}_i \bar{u}_j) = 0, \quad (2.9)$$

where $\bar{\Pi}$ is the momentum flux tensor.

Momentum will be passively convected by the fluid, so we can write part of the momentum flux as $\rho \vec{u} \vec{u}$. Generally, there will also be forces acting on a fluid element, which we can express by adding a force term to the RHS of the momentum equation

$$\frac{\partial (\rho \vec{u})}{\partial t} = -\nabla \cdot (\rho \vec{u} \vec{u}) + \vec{F}. \quad (2.10)$$

Eq. (2.10) is simply Newton's second law applied to a fluid element.

The general force \vec{F} can be broken up into the terms \vec{f} and $\nabla \cdot \vec{\sigma}$,

$$\vec{F} = \vec{f} + \nabla \cdot \vec{\sigma}, \quad (2.11)$$

where $\vec{\sigma}$ represents internal stresses and \vec{f} is an external body force, such as gravity or the Lorentz force (in MHD). The stress tensor represents the force the adjacent fluid exerts on a fluid element Figure 1. The scalar pressure is usually identified with the diagonal elements and the off-diagonal components represent shear stress (or "drag").

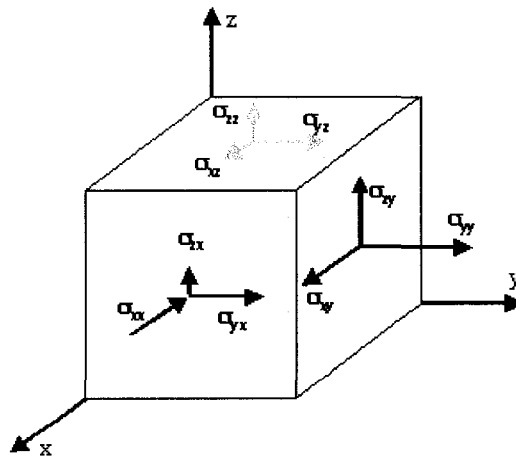


Figure 1: The components of the stress tensor, acting on an infinitesimal fluid element. The diagonal components (normal to the surface elements) are usually identified with the scalar pressure.

Newton observed that, for a wide class of fluids, the shear stress exerted by fluid elements on each other is proportional to the velocity gradient between them.

Accordingly, $\vec{\sigma}$ can be represented by

$$\kappa_{\alpha\beta} \partial \partial \kappa_{\alpha\beta} P \partial \kappa \frac{\partial \partial u_\alpha}{\partial \partial x_\beta} \partial \frac{\partial u_\beta}{\partial x_\alpha} \partial \partial \partial \kappa_{\alpha\beta} P \partial 2 \kappa S_{\alpha\beta}, \quad (2.12)$$

where ρ is the *viscosity*, P is the scalar pressure and $S_{\alpha\beta} = \frac{1}{2} \left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right)$ is called

the *strain-rate tensor*. Fluids that obey the above equation are called *Newtonian fluids*.

Most fluids can be considered Newtonian, although there are important exceptions, e.g., quicksand, blood, etc.

Assuming the stress tensor is Newtonian, the momentum equation reads

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = \frac{1}{\lambda} \nabla P + \lambda^{-2} \nabla \cdot \vec{u} + \lambda^{-1} \nabla \cdot \vec{\sigma}, \quad (2.13)$$

where we have used the continuity equation, $\lambda = \frac{\mu}{\rho}$ is the *kinematic viscosity*, and $\sigma = \frac{\rho \kappa}{\lambda}$

is called a *bulk viscosity*. We note in passing that the last term in Eq. (2.13) will generally be negligible. (See Section 2.4)

Eqs. (2.8) and (2.13) are referred to as the Navier-Stokes equations. The second term on the LHS of Eq. (2.13) is the nonlinear advection term. It is this nonlinearity which gives rise to all of the myriad complex fluid phenomena and makes computational fluid dynamics challenging.

So far we have four equations for five variables (ρ, \bar{u} and P). To close the system we need an expression for the pressure in terms of the other variables. The speed of sound, c_s , is defined as

$$c_s^2 = \left(\frac{\partial P}{\partial \rho} \right)_{\bar{u}}. \quad (2.14)$$

Given the equation of state for an ideal gas,

$$P = \rho \frac{\kappa k T}{m}, \quad (2.15)$$

we see that for fluid at a constant temperature c_s is a constant. An appropriate expression for the isothermal scalar pressure is therefore

$$P = \rho c_s^2. \quad (2.16)$$

Eqs. (2.8), (2.13) and (2.16) and now form a closed nonlinear system of PDE's in the variables ρ, \bar{u} and P .

2.4 Scalings for the Navier-Stokes Equations

In order to identify small terms in the Navier-Stokes equations, it is useful to introduce characteristic length and velocity scales, and to transform to dimensionless units. Let us define

$$\bar{x} = L \tilde{x} \quad (2.17)$$

$$\bar{u} = U \tilde{u}, \quad (2.18)$$

where L and U are characteristic scales and velocities, respectively. Substituting these expressions into the momentum equation gives

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = \frac{1}{\lambda} \left(P - \frac{1}{\text{Re}} \nabla^2 \vec{u} \right), \quad (2.19)$$

where

$$\text{Re} = \frac{UL}{\kappa} \quad (2.20)$$

is called the Reynolds number. The Reynolds number is dimensionless and characterizes the relative importance of nonlinear convection compared to dissipation.

In the limit $\text{Re} \gg 1$, the momentum equation reduces to the dispersion equation, which tends to smear small-scale structures. In the opposite limit ($\text{Re} \ll 1$), the nonlinearity dominates. Thus, high Re flows are typically chaotic, with fluctuations on a wide range of scales – that is, they are turbulent. Fluid flow typically begins the transition from laminar to turbulent at about $\text{Re} \sim 1000$ and is fully turbulent at Re of several thousand.

For the flows considered in this thesis, the characteristic flow speed is low compared to the speed of sound. This condition can be expressed as

$$\text{Ma} = \frac{U}{c_s} \ll 1, \quad (2.21)$$

where Ma is called the Mach number. Many flows of practical interest satisfy the low-Ma condition. For instance, the speed of sound in the lower atmosphere is about 330 m/s, so the movement of air around people, trains and cars are all characterized by a low Ma.

When the Mach number is below about 0.3, it is observed that the density is essentially constant following a fluid element, that is

$$\frac{\partial \kappa}{\partial t} + \bar{u} \cdot \nabla \kappa = 0 \quad (2.22)$$

$$\Omega \cdot \nabla \bar{u} = 0. \quad (2.23)$$

Note that Eq. (2.21) is a condition on the flow, not on the fluid; such flows are called incompressible. Incompressibility is a very useful simplifying assumption – for an incompressible flow the last term in Eq. (2.13) can be neglected. For the remainder of this thesis we will work to $O(Ma^2)$.

Note that Eq. (2.23) does not necessarily mean that the density is constant as many authors imply, but rather that its value does not change following a fluid element. We will, however, consider nearly uniform densities. For an initially uniform density field, we adopt the ordering that the density (and thus pressure) variations are of $O(Ma^2)$ [2]

$$\rho \sim \rho_0 + \sigma \sim O(Ma^2) \quad (2.24)$$

so ρ can be freely moved inside the derivatives in Eqs. (2.13) and (2.22). Thus, a convenient form of the Navier-Stokes equations for our purposes is

$$\frac{\partial \kappa}{\partial t} + \bar{u} \cdot \nabla \kappa = 0 \quad (2.25)$$

$$\frac{\partial \kappa \bar{u}}{\partial t} + \bar{u} \cdot \nabla (\kappa \bar{u}) + \nabla \cdot (\kappa \bar{u} \bar{u}) = \nabla \cdot (\kappa \bar{\tau}) \quad (2.26)$$

where \bar{I} is the identity matrix. Note that, except for the viscous term, these equations are in conservation form, which is often advantageous when formulating numerical

methods. The use of these equations to mimic Eqs. (2.13) and (2.22) is called a *quasi-incompressible* formulation.

2.5 Phenomenology of Turbulence

Turbulence is ubiquitous in nature. The water coming out of the sink, the airflow around a car, and the breath flowing in and out of your lungs right now are all turbulent. The kinematic viscosities of water and air in m^2/s are $O(10^{-6})$ and $O(10^{-5})$, respectively, so the Reynolds number of all but the slowest/smallest flows will be large.

A precise definition of turbulence is lacking, but there are certain characteristics that are generally implied when a flow is called “turbulent”:

1. Irregular fluctuations of the velocity field on a wide range of spatial- and time-scales (a statistical approach is necessary);
2. Three-dimensionality (two-dimensional “turbulence” has qualitatively different behavior);
3. Dissipative (requires a continuous input of energy); and
4. High Reynolds number (dissipation is small compared to advection).

In many of the simulations described in this dissertation, the initial conditions are set up to contain large-scale gradients which evolve into smaller structures. There is no energy input, so dissipation tends to damp out field gradients over time. This is

referred to as freely decaying turbulence and allows for reproducible systems that one can use as numerical benchmarks.

For geometries of relevance to engineers, following the details of all of the structures present in a turbulent flow is both impossible and unnecessary. Different realizations of the same turbulent flow usually have similar statistical properties. It is therefore the hope that a general theory of turbulence can be discovered that will describe the statistical properties of turbulent hydrodynamic fields. Despite over 150 years of intense work by thousands of physicists, engineers and mathematicians, no such theory has been found. This has led some to call turbulence one of “the great unsolved problems of classical physics.”

2.6 Kolmogorov Theory

In the 1940's Andrei N. Kolmogorov introduced a number of scaling laws that represent one of the first successful attempts to quantify the structure of fully developed isotropic turbulence. We shall see that energy flow occurs locally in k -space – that is, eddies tend to transfer their energy to other eddies of a slightly smaller size. Kolmogorov envisioned a process whereby energy is injected into the system at low- k and cascades to higher and higher k -numbers until it is exponentially damped at the small dissipative scale.

We can obtain an evolution equation for the kinetic energy by dotting \vec{u} into Eq. (2.13) and integrating over all space,

$$\frac{\partial}{\partial t} \int \frac{1}{2} u^2 d^3x - \int \frac{1}{2} \bar{u} (\nabla^2 d^3x) - \int \frac{1}{2\lambda} \bar{u} (\nabla \cdot P d^3x) - \int \frac{1}{2} \bar{u} (\nabla^2 \bar{u}) d^3x. \quad (2.27)$$

If we assume that \bar{u} vanishes at infinity we can integrate all the terms containing a spatial derivative by parts to obtain,

$$\frac{\partial}{\partial t} \int \frac{1}{2} u^2 d^3x - \int \frac{1}{2} \bar{u} (\nabla^2 d^3x) - \int \frac{1}{2} (\partial \bar{u})^2 d^3x, \quad (2.28)$$

where we have used $\Omega \bar{u} \Omega = 0$. We see that the nonlinear term does not change the energy, but rather merely shuffles it to different scales. It is solely the viscous term which is responsible for dissipating the kinetic energy.

It is assumed that far from boundaries, the nature of structures which are smaller than the scale at which energy is injected into the system, but larger than the dissipation scale, is universal and self-similar (and therefore does not depend on the viscosity). If energy could build up at any particular wave-number, then structures of that size would dominate the energy spectrum. We conclude that in this so-called *inertial sub-range* kinetic energy is merely “passing through” on its way to being damped at small scales.

By hypothesis then, the energy per unit mass at a given wavenumber (that is, the energy spectrum) depends only on the viscous energy flux, ε , and the wavenumber k ,

$$E(k) \propto f(\rho, k). \quad (2.29)$$

The units of these quantities are

$$\begin{aligned} k &\sim \text{length}^{-1} \\ \text{Energy (per mass)} &\sim \text{length}^2 / \text{time}^2 \\ \text{Energy spectrum} &\sim \text{length}^3 / \text{time}^2 \\ \text{Energy flux} &\sim \text{Energy/time} \sim \text{length}^2 / \text{time}^3 \end{aligned}$$

By a unit analysis, we see that the energy spectrum scales as

$$E(k) = C\varepsilon^{2/3}k^{-5/3}, \quad (2.30)$$

where the proportionality constant C is known from experiment to be of order 1 [3].

This important result is known as the Kolmogorov-5/3 law and is depicted in Figure 2.

It is often used in experiments and simulations to test whether turbulence is fully developed.

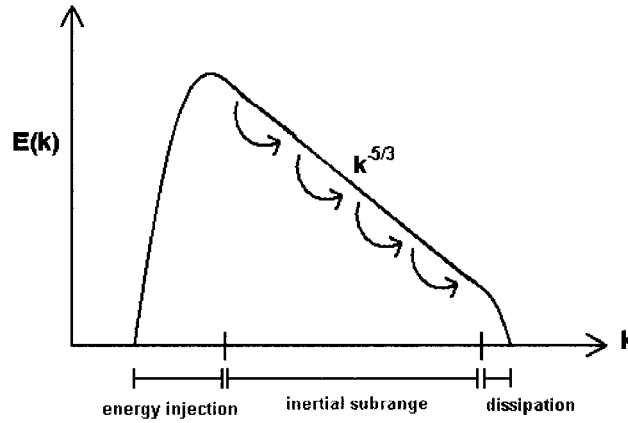


Figure 2: A stylized depiction of the cascade of kinetic energy in wavenumber space for isotropic turbulence. Dissipation becomes significant at the scale $k_{Kol} \sim 1/L_{Kol}$.

To estimate the size of the smallest eddies, we note that Eq. (2.28) can be used to relate the viscous flux to the viscosity and the velocity derivatives at the small scale,

$$\nu \sim \nu(U_{Kol}^2 / L_{Kol}^2), \quad (2.31)$$

where U_{Kol} and L_{Kol} are velocity and size of the smallest eddies. L_{Kol} is sometimes called the Kolmogorov length.

The energy dissipation ν must be balanced by the rate at which energy is injected into the large scales,

$$K \sim U^3 / L, \quad (2.32)$$

where U and L are the characteristic velocity and length of the system. Lastly, we note that at the dissipative scale,

$$U_{Kol} L_{Kol} \sim \lambda. \quad (2.33)$$

Combining Eqs. (2.31), (2.32) and (2.33) we obtain the scaling for the size of the smallest flow structures,

$$L_{Kol} \sim L \text{Re}^{3/4}. \quad (2.34)$$

This equation is very useful in CFD, since it allows one to estimate the amount of resolution needed for a large production run.

The arguments in this section involve more than a little hand waving. We have not actually defined what is meant by “eddy,” relying instead on the intuition that comes from dealing with fluids on a daily basis. Nevertheless, the expressions presented here conform quite well to experiment, and represent the most useful attempt to date in trying to quantify turbulence.

Chapter 3

Numerical Solutions of Partial Differential Equations

Nonlinear partial differential equations are generally hard to solve. In fact, systems of coupled nonlinear partial differential equations can only be solved analytically in very special cases (e.g., soliton physics), or when the system under consideration has a very high degree of symmetry. Most systems of practical interest will involve complicated boundary conditions, so numerical techniques are required.

In many fluid flow problems of engineering interest, the transport coefficients will be quite small, but nonzero. As elaborated upon in Section 2.4, when transport coefficients are small we expect convection processes to dominate over diffusion and allow very small scale structures to develop. Due to the huge number of degrees of freedom of such systems, CFD and computational plasma physics are some of the most numerically demanding computer applications.

3.1 Parallel Computing

Electronic computers have existed for approximately seven decades. It was noted by Gordon Moore in 1965 that, due to advances in microchip fabrication, the

number of transistors on integrated circuits roughly doubles every two years, an empirical result known as Moore's Law. This "law" implies that computing power grows at an exponential rate. As Moore himself noted, for a variety of reasons, such a growth rate cannot be sustained indefinitely, and will eventually reach certain fundamental physical limits. Transistors are already of the order of tens of nm in length. At this scale, heat-induced failure becomes a serious issue. Unless (or until) quantum computers become a practical alternative, the future of high performance computing lies not with continued miniaturization, but with parallel computing.

A parallel computer is a machine with multiple processors working on the same problem simultaneously and communicating via a network. Many parallel architectures exist, distinguished by how the processors are connected and how they share memory. In a fluid problem, a common way to parallelize is to divide the domain up into a number of sub-domains, with each process computing the flow for its sub-domain. Of course, when fluid flows from one sub-domain to another, the various processes must communicate with each other.

A good parallel algorithm is scalable or *efficient*, meaning that the speed-up of the execution time is proportional to the number of processors employed. If the computing power is proportional to the number of processors used, the efficiency is 100%. For instance, computing the value of a well-behaved definite integral is a highly parallelizable operation (called *embarrassingly parallel*), since the integral across any given sub-domain can be computed independently of the others, with the results summed at the end.

Many common CFD algorithms involve inherently non-local computations. In the finite element method (Section 3.3) for instance, one must solve an $N \times N$ linear system at every time step, where N is the number of nodes in the simulation. Because of the non-local nature of such an operation, the scalability of such algorithms is limited. This is often manifested by diminishing parallel efficiency as the number of processors is increased, as shown in Figure 3. A great deal of work has gone into various tricks to increase the efficiency and scalability of parallel algorithms, but it is important to bear in mind that this phenomenon is inherent to certain algorithms.

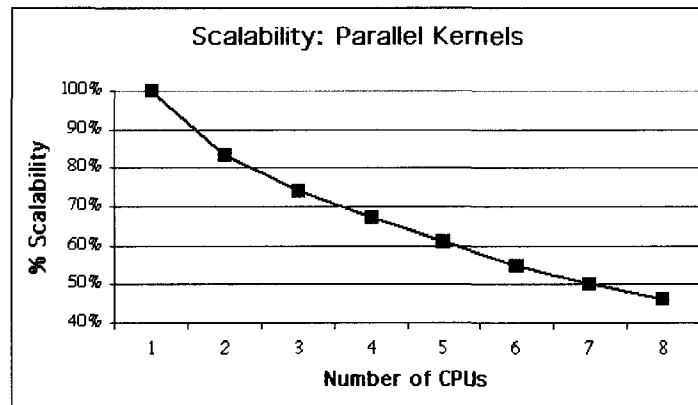


Figure 3: *The efficiency of an LU linear solver as the number of processors is increased. From www.sun.com.*

As we shall see in Section 4.2, the most commonly used form of the LBM is remarkably parallelizable. This is in fact one of its strongest attributes. When extending and/or stabilizing the algorithm, it therefore behooves one to do so in such a way that the parallelism of the method is not affected.

3.2 Numerical Stability

In addition to being parallelizable, a good numerical algorithm must be *stable*. Real computers can only store a finite number of decimal places; representing floating point quantities on a real machine means that round-off errors are unavoidable. Often an algorithm iterates a certain computation many times. If an algorithm amplifies the round-off errors to the extent that they quickly swamp the true answer, then it is termed *unstable*.

On a hypothetical infinite-precision (but discrete) computer, numerical instabilities would not exist, but in the real world unstable algorithms are practically useless. Generally, an unstable algorithm that “works” on paper will function correctly for certain values of the parameters, but if certain parameters (e.g., transport coefficients) are made too small, then unphysical oscillations will develop and render the solution meaningless.

Numerical instabilities are common in CFD, and are perhaps the biggest stumbling block to the effective utilization of Lattice Boltzmann method for large-scale fluid simulations. The bulk of this dissertation is comprised of various ways to suppress numerical instabilities in the Lattice Boltzmann method over a wide range of parameters.

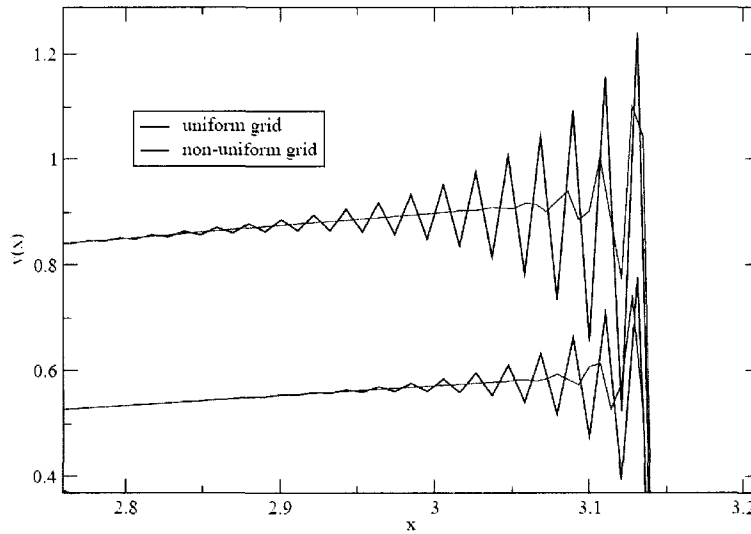


Figure 4: *A close-up of spurious oscillations in the vicinity of a sharp one-dimensional gradient (shock) from [4]. Note that with the non-uniform grid, finer spatial resolution reduces the oscillations. If the parameters are particularly unfavorable, such oscillations can grow indefinitely, rendering the simulation unstable.*

3.3 A Juxtaposition of Methods of Discretization

The fundamental consideration in CFD is how to represent a continuous quantity on a discrete computer. There are a wide variety of methods for discretizing the governing equations and/or their solution.

Finite Differences (FD)

Finite differencing is the most obvious way to discretize a partial differential equation. It is essentially differential calculus in reverse – the equation is approximated by replacing continuous derivatives with finite differences, e.g.,

$$\frac{\partial f}{\partial x} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}. \quad (3.1)$$

There are a large number of ways, called *stencils*, to discretize any given derivative, all of which reduce to the correct expression in the limit $\Delta x \rightarrow 0$. The particular form used depends on the equations being approximated, and on the required stability and accuracy.

In their most basic form, finite difference methods require a uniform grid, though coordinate transformations can be employed to handle somewhat more complicated geometries. The biggest advantage of FD is the simplicity of implementation. Note that with this method, the solution is only obtained at grid points. Of course, virtually all fluid simulations need to perform some sort of finite differencing on the time coordinate.

Spectral Methods

Spectral methods involve solving a set of equations in a Fourier transformed wavenumber space. In *pseudo-spectral* methods only some of the terms are Fourier transformed. The major advantage of such methods is exponential accuracy as the number of simulated Fourier modes increases. However, imposing complex boundary conditions, which could be introduced e.g., by the presence of obstacles or a toroidal geometry, is very difficult.

Thus, spectral-type methods are largely restricted to working in periodic or simply-shaped domains. Periodic boundaries can be useful for understanding the behavior of a fluid in the bulk (far from physical boundaries), or for testing a numerical method. Spectral methods are therefore often used to produce numerical

benchmarks and to examine turbulence on small scales, where the dynamics are presumed to be universal.

Finite Volume Method

The finite volume method exploits the divergence theorem to write conservation equations in terms of the flux across surfaces. First, the domain is divided into a number of possibly non-uniform *cells*. As in the finite difference method, the fluid quantities are defined at discrete points, but these discrete values are now identified with the average of the quantity over its cell.

The basic idea can be illustrated by considering the generic conservation law

$$\frac{\partial A}{\partial t} + \nabla \cdot \vec{B}(A) = 0. \quad (3.2)$$

Indexing the cells by i , denoting the volume of the i th cell by V_i and its boundary S_i , we can take the volume average of Eq. (3.2) over a cell,

$$\frac{\partial \bar{A}_i}{\partial t} = \frac{1}{V_i} \int_{V_i} \frac{\partial A}{\partial t} dV = \frac{\partial \bar{A}_i}{\partial t} = \frac{1}{V_i} \oint_{S_i} \vec{B}(A) \cdot d\vec{S} = 0, \quad (3.3)$$

where

$$\bar{A}_i = \frac{1}{V_i} \int_{V_i} A dV \quad (3.4)$$

is the volume average value of A . Note that the time coordinate must still be discretized, often via a simple finite difference. The various finite-volume schemes [5] are distinguished by how they calculate the value of the fluxes at the boundary.

Finite-volume methods are used extensively in the CFD community because, in contrast to finite differences, they allow for the use of a non-uniform mesh so the

resolution can be increased in regions of interest. The method also allows for the resolution of discontinuities in the solution (shocks). A number of schemes can be used to restrict the flux values to physically realistic values, and to prevent the appearance of spurious oscillations in regions of large gradients.

Finite Element Method (FEM)

This method [6] is used extensively in structural analysis, but can also be used for fluid flow. It seems to be the method of choice for large-scale MHD simulations.

First, the problem is reformulated in a variational form. The domain is divided into a number of elements, each of which contains a number of nodes. Across each element the solution is assumed to have a specific functional form, e.g., linear or quadratic in the spatial variables. The solution across the whole domain is uniquely determined by the functional values at the nodes. This results in a large linear system for the nodal values which must be solved at every time step.

Mathematically, this is equivalent to expanding the solution in basis functions with compact support, see Figure 5. One can increase the degree of the polynomial approximation across each element and decrease the number of elements. Such an approach combines characteristics of the finite element method and spectral method and is sometimes called a *spectral element* method.

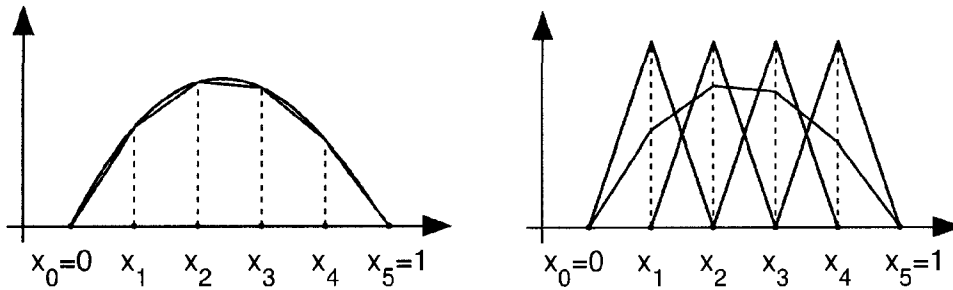


Figure 5: *Left: A linear finite element approximation (red) to the exact solution (blue). Right: A linear combination of the basis functions (blue) is used to construct the finite element solution.*

By using basis functions with small support, one ensures that the matrices that need to be inverted are block diagonal. This enables certain tricks to be used to “precondition” the matrix, and hence speed up the inversion. Nonetheless, matrix inversion is an inherently non-local operation, and as such is not very amenable to parallelization. A major advantage of the FEM is the ease with which it can handle complex boundaries.

Chapter 4

The Lattice Boltzmann Method

4.1 Continuous Kinetic Theory

A typical macroscopic volume of fluid contains on the order of 10^{23} particles. Even if we assume simple classical interactions, this is far too many to ever be simulated on a machine. Furthermore, if we are interested in macroscopic phenomena, we generally don't care about the details of individual particle motion. Thus, a statistical approach – kinetic theory – is needed.

One of the basic quantities of interest in kinetic theory is the single particle distribution function, $f(\vec{x}, \vec{c}, t)$, which describes the probability that a particle will be in an infinitesimal region of phase space around \vec{x} , with speed \vec{c} at time t , irrespective of the position and velocity of other particles. If we assume that particles are neither created nor destroyed and that motion is local in x - and c -space, then the evolution of f is given by the chain rule,

$$\frac{df(\vec{x}, \vec{c}, t)}{dt} = \frac{\partial f}{\partial t} + \vec{c} \cdot \nabla_{\vec{x}} f + \frac{\vec{F}}{m} \cdot \nabla_{\vec{c}} f = 0, \quad (4.1)$$

where $\nabla_{\vec{c}} = \sum_{\alpha} \frac{\partial}{\partial c_{\alpha}}$ is the gradient operator in velocity space. However, in actual fact,

particles can be considered to undergo collisions, which introduce the possibility of non-local jumps in velocity space. The effects of such collisions are represented by

adding a (usually nonlinear) collision term to the right hand side of Eq. (4.1). For now, we will leave the collision operator unspecified and denote it by $\Omega[f]$.

Observable macroscopic quantities can be obtained by multiplying f by various powers of \vec{c} and integrating over all velocities. This is referred to as taking velocity space moments of f , e.g.,

$$\rho(x) \rightarrow \int f(\vec{x}, \vec{c}) d\vec{c} \rightarrow \text{zero moment} \quad (4.2)$$

$$\vec{u} \rightarrow \frac{1}{\rho} \int f(\vec{x}, \vec{c}) \vec{c} d\vec{c} \quad \text{first moment,} \quad (4.3)$$

and possibly higher moments, where ρ and \vec{u} are typically identified with the fluid density and velocity. The information contained in f is essentially equivalent to the infinite hierarchy of the moments of f .

Ludwig Boltzmann introduced a collision operator under the assumptions of binary collisions and molecular chaos (the pre-collision momenta of colliding particles are uncorrelated). Under these assumptions one can show that the functional

$$H(f) \rightarrow \int f \ln(f) d\vec{c} \quad (4.4)$$

always decreases with time, that is,

$$\frac{dH}{dt} \leq 0. \quad (4.5)$$

This is the celebrated H-theorem.

An equilibrium distribution function can be obtained by minimizing H while requiring that the mass and momentum be conserved. This is done by standard Lagrange multiplier methods, yielding the Maxwell-Boltzmann equilibrium distribution function

$$f^{MB}(\vec{x}, \vec{c}) = \frac{m^{-3/2}}{2\epsilon kT} \exp\left[-\frac{mc^2}{2kT}\right], \quad (4.6)$$

where m is the mass of the particles, k is Boltzmann's constant, and T is the temperature.

The full Boltzmann collision operator is nonlinear and involves integrals over the momentum space of two particles, and is impractical for large-scale fluid computations. We would like to find a linear collision operator that can reproduce the correct fluid scale behavior without worrying about the unimportant details of the microscopic dynamics.

A stylized depiction of the evolution of a distribution function under the influence of collisions is shown in Figure 6.

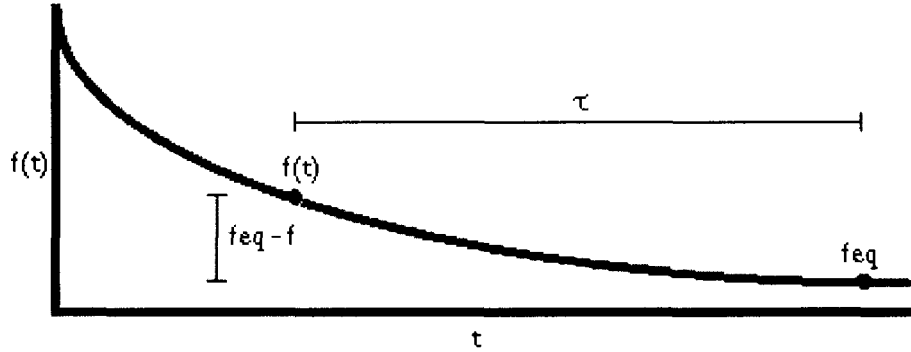


Figure 6: The evolution of the distribution function, suitably averaged over a small region of phase space. The distribution function relaxes at a characteristic time scale ν towards equilibrium.

This plot suggests that, close to equilibrium, we can consider the first order time difference

$$-_{BGK}[f] \left[-\frac{f^{eq} - f}{\epsilon} \right], \quad (4.7)$$

where τ is a characteristic time at which f relaxes towards equilibrium. Eq. (4.7) is known as the Bhatnagar-Gross-Krook (BGK) collision operator [7].

4.2 Lattice Boltzmann

Although historically the Lattice Boltzmann Method was an outgrowth of Lattice Gas Automata [1], it will be advantageous for our purposes to take Eq. (4.1) as our starting point. We will consider the case with no body forces, so we have

$$\frac{\partial f}{\partial t} + \vec{c} \cdot \nabla f = -\frac{f - f^{\text{eq}}}{\tau}. \quad (4.8)$$

This equation must be discretized in the variables \vec{c} , \vec{x} and t . The time variable is almost always handled with a simple first order finite difference as in Eq. (3.1). The discretization of the position and velocity variables can in principle be handled with any of the techniques listed in Section 3.3. For many purposes those discretization techniques retain an unnecessary amount of information and are impractical in a six-dimensional phase space.

The basic idea of the LBM is to discretize in velocity space as drastically as possible. That is, we restrict the continuous variable \vec{c} to the discrete values $\{\vec{c}_i\}$, $i = 1 \dots q$. These discrete *lattice velocities*, $\{\vec{c}_i\}$ are chosen to possess only a few elements, while retaining enough isotropy so that the fluid equations are recovered in the macroscopic limit. The various admissible sets of lattice velocities are referred to as LB models, and are denoted DdQq where d is the number of dimensions and q is

the number of velocities. The velocities of the common LB models D1Q3, D2Q9, D3Q15, D3Q19 and D3Q27 are listed in Table I.

We will denote the corresponding discrete-velocity distribution functions (or populations) $f(\vec{x}, \vec{c}_i, t) \int f_i(\vec{x}, t)$. With these definitions, the single equation in \vec{x}, \vec{c}, t [Eq. (4.8)] becomes q equations in \vec{x}, t

$$\frac{df_i}{dt} \leq \vec{c}_i \leq f_i \leq \leq \} f_j \}. \quad (4.9)$$

Note that, in general, the collision operator will depend on all of the discrete populations. Eq. (4.9) is often called the discrete velocity Boltzmann equation (DVBE).

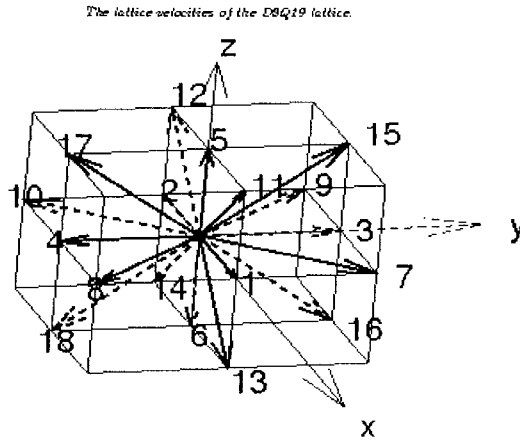


Figure 7: *The lattice vectors of the popular D3Q19 model.*

The integral moments of Eqs. (4.2) and (4.3) are replaced by sums over the discrete velocities,

$$\tau(\vec{x}) \sum_{i \in \Sigma} f_i(\vec{x}) \quad (4.10)$$

$$u(\bar{x}) = \frac{1}{\tau(\bar{x})} \sum_{i \in \Sigma} \bar{c}_i f_i(\bar{x}). \quad (4.11)$$

A particularly convenient formulation can be constructed by discretizing Eq. (4.9) in space and time using simple forward differencing,

$$\frac{f_i(x_\tau, t + \Delta t) - f_i(x_\tau)}{\Delta t} = \sum_{i, \tau} c_{i, \tau} \frac{f_i(x_\tau + \sum x_\tau, t + \Delta t) - f_i(x_\tau, t)}{\sum x_\tau} \sum_{i, \tau} \{f_i(x_\tau, t)\}, \quad (4.12)$$

Model	Lattice Velocities (plus permutations)	Speeds (# of corresponding velocities)
D1Q3	-1, 0, 1	0(1), 1(2)
D2Q9	0, 0, ±1, ±1, ±1, ±1	0(1), 1(4), √2(4)
FD3Q15	±0, 0, 0, ±1, ±1, 0, 0, ±1, ±1, ±1, ±1	0(1), 1(6), √3(8)
D3Q19	±0, 0, 0, ±1, ±1, 0, 0, ±1, ±1, 0, 0	0(1), 1(6), √2(12)
D3Q27	±0, 0, 0, ±1, ±1, 0, 0, ±1, ±1, 0, 0, ±1, ±1, ±1, ±1	0(1), 1(6), √2(12), √3(8)

TABLE I. The lattice velocities and speeds of common LB models.

where we have evaluated the collision operator at the old time step (explicitly), but the convective term at the new time step (implicitly). If we couple the velocity-space

lattice to the spatial grid such that $c_{i, \tau} = \frac{\pm x_{i, \tau}}{\pm t}$ then Eq. (4.12) reduces to the simple

form

$$f_i(\bar{x} + \bar{c}_i \Delta t, t + \Delta t) = f_i(\bar{x}, t) + \Delta t \sum_{i, \tau} \{f_i(\bar{x}, t)\}. \quad (4.13)$$

This equation is often called the lattice Boltzmann equation (LBE). With the usual choice of Eq. (4.7) we have

$$f_i(\mathbf{x}, t + \Delta t) = f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t) - \frac{\Delta t}{\tau} (f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) \quad (4.14)$$

This is referred to as the lattice BGK (LBGK) equation.

Taking Eq. (4.14) as the basis of a numerical procedure produces an exceptionally simple and parallelizable algorithm. Essentially there are two steps: the streaming (depicted in Figure 8), which involves no computations, and the collision, which involves only local information (i.e., the populations at a single spatial node). The standard LB algorithm which implements Eq. (4.14) is depicted schematically in Figure 9.

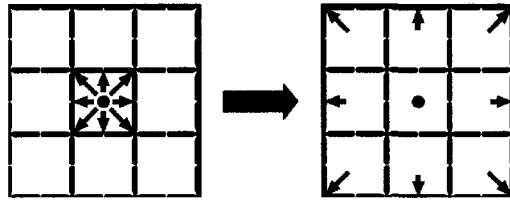


Figure 8. During the streaming process, the populations are shifted unchanged to adjacent nodes.

The locality of the collision process enables the standard LBM to be parallelized extremely efficiently. Given all the populations in a sub-domain of coordinate space, a processor can compute the hydrodynamic moments, the equilibrium distribution function and the post-collision populations, \tilde{f} , all without the need to communicate with the other processes. It is only in the streaming step that the processors need to communicate. Thus, in the LBM, the scaling of the number of floating point operations per second (FLOPS) with the number of processors is almost

ideal, as shown in Figure 10. This scalability is considered one of the primary advantages to discretizing the Boltzmann equation as in Eq. (4.14).

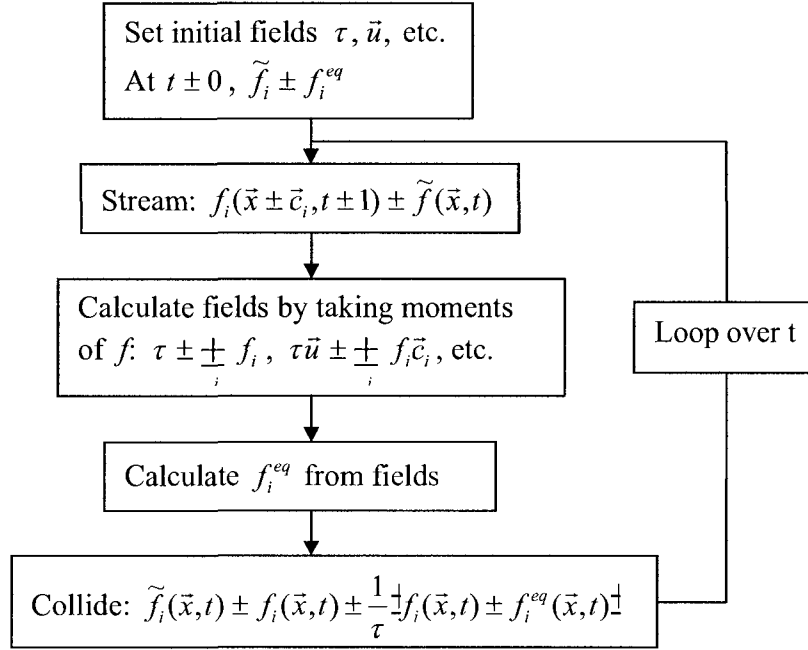
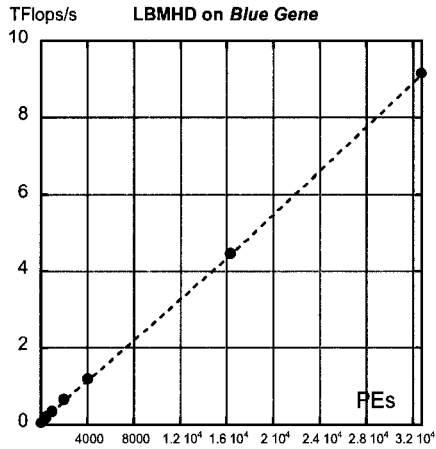
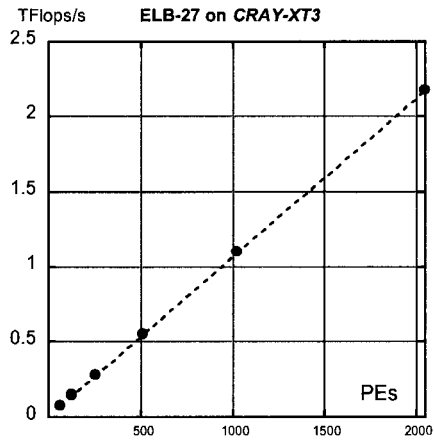


Figure 9. A pictorial representation of the basic LB algorithm. \tilde{f} is the post-collision distribution function. If non-periodic boundary conditions are to be used, they are implemented after streaming.

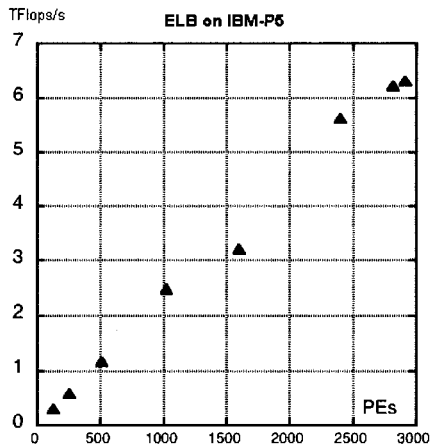
Because the velocity space is so drastically truncated, much of the original kinetic (velocity distribution) information is lost. Thus, the LBM should be considered an alternative method for computing fluid flow, rather than a true kinetic method. The major advantage of working at the mesoscopic level, rather than the fluid level is that the nonlinear convective derivatives of fluid-level equations have been replaced by the simple linear constant advection of the DVBE. The tradeoff is the larger number of evolved quantities at the mesoscopic level (q populations) as compared to the fluid level (usually τ, \vec{u} , sometimes T).



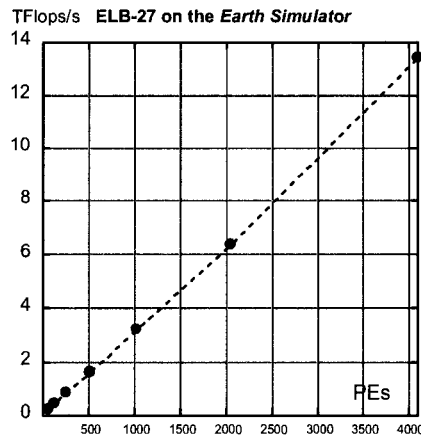
(a) 32768 PEs, 9.1 TFlops/s, 23% peak



(b) 2912 PEs, 6.3 TFlops/s, 30% peak



(c) 2048 PEs, 2.2 TFlops/s, 45% peak



(d) 4084 PEs, 13.5 TFlops/s, 41% peak

Figure 10. The scaling of the computing power of a LB code with the number of processors used on several supercomputer architectures. A straight line corresponds to 100% efficiency. Note that not only is the scaling almost perfect, but the FLOP-rate itself is quite high.

It is worth emphasizing that the way of discretizing Eq. (4.9) as presented above is not the only way. Eq. (4.14) is generally called “the” LB equation because it historically evolved from LGA, and because of its superior parallel performance.

The LBM is perhaps best viewed as an original paradigm for solving fluid-type equations, rather than as a specific algorithm. Taking Eq. (4.9) as the basis of the LBM, rather than Eq. (4.14), enables one to use the full repertoire of CFD in its solution. Other authors have used the finite volume method [8], finite element method [9], characteristics-based methods [10], and a variety of finite difference discretizations to solve Eq. (4.9) [11]. In this dissertation I will focus on Eq. (4.14) due to its simplicity and near perfect parallelization.

4.3 The Chapman-Enskog Procedure

All of the physics contained in the LBGK equation is in the equilibrium distribution. In order to select the equilibrium that will correctly model the macroscopic equations, we need a connection between the kinetic and fluid levels. This connection is provided by an asymptotic multi-scale expansion called the Chapman-Enskog procedure. The Chapman-Enskog procedure for continuous kinetic theory with the full Boltzmann collision operator is quite mathematically involved [12]. Using the BGK operator simplifies the analysis considerably [13]. Here I shall perform the procedure on the LBGK equation, Eq. (4.14). This treatment follows [4].

We begin by expanding $f_i(\vec{x}, \vec{c}_i, t)$ in the small parameters ϵ and τ ,

$$f_i(\vec{x}, \vec{c}_i, t) = \sum_{n=0}^{\infty} \frac{\epsilon^n \tau^n}{n!} f_i^{(n)}(\vec{x}, \vec{c}_i, t) \quad (4.15)$$

and substituting this into Eq. (4.14) to get

$$\sum_{n=0}^{\infty} \frac{\alpha^n}{n!} \partial_{x_i}^n c_{i,\tau} f_i(\bar{x}, t) \approx \frac{1}{\tau} \partial_{x_i} f_i^{eq} \quad (4.16)$$

Note that the lower index has been changed due to the cancellation of the $n = 0$ term.

The Chapman-Enskog procedure is based on the observation that there are two disparate timescales in most fluid-type problems. When we consider putting cream into coffee, for instance, we notice that mixing occurs much more quickly if it is stirred as opposed to waiting for diffusion to take place. We will treat the faster convective timescale, t_0 , and the slower resistive or transport timescale, t_1 , as formally independent variables. Accordingly, we introduce a small parameter ε and the orderings

$$\partial_{x_i} \approx \varepsilon \partial_{x_{i0}} \approx \varepsilon^2 \partial_{x_{i1}} \approx \dots \quad (4.17)$$

$$t \approx \varepsilon t \quad (4.18)$$

$$f_i \approx f_i^{(0)} \approx \varepsilon f_i^{(1)} \approx \varepsilon^2 f_i^{(2)} \approx \dots \quad (4.19)$$

ε can be identified with the Knudsen number or the Mach number.

To lowest order, it is immediately seen that $f_i^{(0)} \approx f_i^{eq}$. Inserting Eqs. (4.17), (4.18) and (4.19) into Eq. (4.14) and collecting terms of order ε gives

$$\partial_{x_{i0}} f_i^{(0)} \approx \partial_{x_{i,\varepsilon}} c_{i,\varepsilon} f_i^{(0)} \approx \frac{1}{\varepsilon} f_i^{(1)} \quad (4.20)$$

where the (constant) \bar{c}_i have been moved inside the derivatives. To order ε^2 , the LBGK equation is

$$\partial_{x_{i1}} f_i^{(0)} \approx \partial_{x_{i0}} \partial_{x_{i,\varepsilon}} c_{i,\varepsilon} f_i^{(1)} \approx \frac{1}{2} \partial_{x_{i0}} \partial_{x_{i,\varepsilon}}^2 f_i^{(0)} \approx \frac{1}{\varepsilon} f_i^{(2)}. \quad (4.21)$$

It will sometimes be convenient to use Eq. (4.20) to eliminate $f_i^{(1)}$ from Eq. (4.21),

$$\partial_t f_i^{(0)} - \frac{1}{2} \partial_{i_0}^2 f_i^{(0)} - \partial_{i_0} c_{i_0} f_i^{(0)} = \frac{1}{\varepsilon} f_i^{(2)}. \quad (4.22)$$

Taking the moments of Eq. (4.20) will give the ideal-level fluid equations. The moments of Eq. (4.22) introduce the possibility of dissipation (e.g., viscosity). Going to higher order in ε would involve higher spatial derivatives; however, these terms become very small quickly. Eqs. (4.20) and (4.22) will suffice for our purposes.

4.4 Lattice Boltzmann for the Navier-Stokes Equations

To complete the process of recovering a system of fluid equations, we need to take the moments of Eqs. (4.20) and (4.22). We will first consider the ideal level, Eq. (4.20). Conservation of number density and momentum require the solvability conditions

$$\sum_{i=0}^q f_i^{(n)} = 0, \text{ and} \quad (4.23)$$

$$\sum_{i=0}^q \vec{c}_i f_i^{(n)} = 0 \quad (4.24)$$

for integer $n \geq 1$. Summing Eq. (4.20) over i gives

$$\frac{\partial \varepsilon}{\partial t} = \varepsilon \vec{\nabla} \cdot \vec{u} \geq 0, \quad (4.25)$$

where

$$\varepsilon \geq \sum_i f_i \geq \sum_i f_i^{eq} \text{ and} \quad (4.26)$$

$$\varepsilon \vec{u} \propto \prod_{i \in \Omega} \bar{c}_i^q f_i \propto \prod_{i \in \Omega} \bar{c}_i^q f_i^{eq}. \quad (4.27)$$

Multiplying by \bar{c}_i and then summing over i gives

$$\frac{\alpha \varepsilon \vec{u}}{\alpha} \propto \bar{\omega}^{(0)}, \quad (4.28)$$

where

$$\bar{\omega}^{(n)} \propto \prod_{i \in \Omega} \bar{c}_i^q \bar{c}_i f_i^{(n)}. \quad (4.29)$$

for integer n .

One can continue to take higher moments of the kinetic equations if temperature/pressure equations are desired. Each equation will couple the i th moment to the $i+1$ moment, so a closure is needed. A closure usually must take into account thermodynamic considerations, and reflects the nature of the propagating medium. Thermal LB exhibits a number of troubling defects, notably numerical instability outside of a narrow parameter range [14], so the LBM is ordinarily used for isothermal flows with the equation of state given by Eq. (2.16). Therefore the second moment of the equilibrium should be

$$\vec{\Pi}^{(0)} \propto \varepsilon c_s^2 \vec{I} \propto \varepsilon \vec{u} \vec{u}, \quad (4.30)$$

where \vec{I} is the identity matrix.

Since the velocities under consideration are small in lattice units, it suffices to consider equilibria that are polynomials in \vec{u} . A particularly convenient formulation which gives the correct moments [Eqs. (4.26), (4.27) and (4.30)] for the athermal Navier-Stokes equations is

$$f_i^{eq} = w_i \varepsilon \left[1 + 3\bar{u}_i \bar{u}_i + \frac{9}{2} \bar{u}_i \bar{u}_i + \frac{3}{2} u^2 \right], \quad (4.31)$$

where the weights w_i depend only on the speed of the associated lattice vector. The weights for some common LB models are given in Table II.

Speed	D2Q9	D3Q15	D3Q19	D3Q27
0	$w_0 = \frac{4}{9}$	$w_0 = \frac{2}{9}$	$w_0 = \frac{1}{9}$	$w_0 = \frac{8}{27}$
1	$w_1 = \frac{1}{9}$	$w_1 = \frac{1}{9}$	$w_1 = \frac{1}{18}$	$w_1 = \frac{2}{27}$
$\sqrt{2}$	$w_2 = \frac{1}{36}$	-	$w_2 = \frac{1}{36}$	$w_2 = \frac{1}{54}$
$\sqrt{3}$	-	$w_3 = \frac{1}{72}$	-	$w_3 = \frac{1}{256}$

TABLE II. The weights that appear in the polynomial equilibriums for common two- and three-dimensional LB models.

Eq. (4.31) is not the only expression that gives the correct moments, but it has favorable stability characteristics and is the low- Ma expansion of the Maxwell-Boltzmann equilibrium [Eq. (4.6)].

To introduce non-ideal effects, we must take the moments of Eq. (4.22).

Summing Eq. (4.22) over i gives

$$\int_{t_0}^t \varepsilon \left[\varepsilon + \frac{1}{2} \int_{t_0}^t \varepsilon^2 \left[2 \int_{t_0}^t \varepsilon \left[\varepsilon u_\varepsilon \right] + \int_{t_0}^t \int_{t_0}^t \varepsilon \beta \right] \right] dt = 0 \quad (4.32)$$

$$\int_{t_0}^t \varepsilon \left[\varepsilon + \frac{1}{2} \int_{t_0}^t \int_{t_0}^t \varepsilon \left[\varepsilon u_\varepsilon \right] \right] dt = \int_{t_0}^t \int_{t_0}^t \int_{t_0}^t \varepsilon \left[\varepsilon \int_{t_0}^t \int_{t_0}^t \varepsilon u_\varepsilon + \int_{t_0}^t \int_{t_0}^t \varepsilon \beta \right] dt = 0. \quad (4.33)$$

Using the ideal equations, Eqs. (4.25) and (4.28), we see that both terms in brackets vanish, leaving us with

$$\Gamma_{t_1} \varepsilon \Gamma \mathbf{0}, \quad (4.34)$$

so the first time scale does not contribute to the continuity equation.

The resistive level contributions to the momentum equation can be elucidated by multiplying Eq. (4.22) by $c_{i,e}$ and summing over i ,

$$\Gamma_{t_1} \Gamma \gamma u_\gamma \Gamma \left[\Gamma \frac{1}{2} \Gamma \Gamma_{t_0}^2 \Gamma \gamma u_\gamma \Gamma \Gamma \frac{1}{2} \Gamma \Gamma_{t_0} \Gamma_{\beta} \Gamma_{\gamma\beta}^{(0)} \Gamma \Gamma_{\beta} \Gamma_{\gamma} \Gamma_i f_i^{(0)} c_{i,\gamma} c_{i,\beta} c_{i,\gamma} \right] \Gamma \mathbf{0} \quad (4.35)$$

$$\Gamma_{t_1} \Gamma \gamma u_\gamma \Gamma \left[\Gamma \frac{1}{2} \Gamma \Gamma_{t_0} \Gamma_{t_0} \Gamma \gamma u_\gamma \Gamma \Gamma_{\beta} \Gamma_{\gamma\beta}^{(0)} \Gamma \Gamma \frac{1}{2} \Gamma \Gamma_{\beta} \Gamma_{t_0} \Gamma_{\gamma\beta}^{(0)} \Gamma \Gamma_{\beta} \Gamma_{\gamma} \Gamma_i f_i^{(0)} c_{i,\gamma} c_{i,\beta} c_{i,\gamma} \right] \Gamma \mathbf{0} \quad (4.36)$$

The second term vanishes by virtue of Eq. (4.28). Since these entire equations are γ smaller than the ideal level equation, we can drop terms that are $O|Ma^2|$ or smaller.

With the chosen scalings [Eq. (2.24)], it can be shown that the term $\Gamma_{\beta} \Gamma_{t_0} \Gamma_{\gamma\beta}^{(0)}$ is $O|Ma^2|$, and will thus be neglected (see Appendix).

We are left with

$$\Gamma_{t_1} \Gamma \gamma u_\gamma \Gamma \left[\Gamma \frac{1}{2} \Gamma \Gamma_{\beta} \Gamma_{\gamma} \Gamma_i f_i^{(0)} c_{i,\gamma} c_{i,\beta} c_{i,\gamma} \right]. \quad (4.37)$$

The RHS can be evaluated with the help of Eq. (4.31), to give

$$\Gamma_{t_1} \Gamma \gamma u_\gamma \Gamma \left[\frac{1}{3} \Gamma \frac{1}{2} \Gamma_{\beta}^2 u_\gamma \Gamma O|Ma^2| \right], \quad (4.38)$$

where $O|Ma^2|$ represents terms proportional to $\Gamma \vec{u}$.

Adding together Eqs. (4.28) and (4.38) gives the complete evolution equation for the fluid momentum,

$$\frac{\partial \langle \gamma \tilde{u} \rangle}{\partial t} = - \left[\frac{\gamma}{3} \tilde{I} \langle \gamma \tilde{u} \tilde{u} \rangle - \left[\frac{1}{3} \langle \gamma \rangle - \frac{1}{2} \langle \tilde{u}^2 \rangle \right] \right] \tilde{u}, \quad (4.39)$$

which is identical to Eq. (2.26), provided we make the identifications

$$\gamma = \left[\frac{1}{3} \langle \gamma \rangle - \frac{1}{2} \langle \tilde{u}^2 \rangle \right] \text{ and} \quad (4.40)$$

$$c_s = \frac{1}{\sqrt{3}}. \quad (4.41)$$

Note that μ , \tilde{u} and $\tilde{\Gamma}$ were *defined* as the velocity moments of the equilibrium distribution, and, as such, can be given varying interpretation depending on the nature of the equilibrium selected. Of course, for the standard Navier-Stokes model presented here, μ is the mass density, $\mu \tilde{u}$ as momentum density and $\tilde{\Gamma}$ as the pressure tensor. However, these are hardly the sole consistent interpretation of these equations. When used to simulate MHD, the zero moment of the distribution functions can be interpreted as the components of the magnetic field [2]; for the shallow water equations, the zero moment is the water depth [15]. In fact, the LBM is capable of simulating a variety of systems of PDE's, so long as they can be expressed in the conservation form of Eq. (2.6).

The BGK operator is the simplest collision operator to implement, and is used almost exclusively in the literature. It is worth noting that the use of a single time parameter at which all moments relax toward equilibrium implies that all transport coefficients in the simulated (macroscopic) equations are the same. For instance, taking the second moment of the kinetic equations (4.20) and (4.22) gives an equation for the temperature evolution,

$$\frac{\partial T}{\partial t} = \gamma T \nabla^2 T, \quad (4.42)$$

where

$$\gamma T = \int_i f_i \bar{c}_i |\vec{u}|^2. \quad (4.43)$$

Since there is only one relaxation time, we find that the BGK operator requires that heat and momentum diffuse at the same rate, $\kappa = \kappa$. This important limitation on the utility of the BGK operator can be dealt with in several ways [16], [17]. In this dissertation, we will be dealing exclusively with athermal (or isothermal) flows, so the BGK operator will suffice.

Chapter 5

An Implicit Lattice Boltzmann Scheme

As indicated in earlier, the Achilles' heel of the standard LBM is numerical instabilities at low values of the transport coefficients. Much work has gone into stabilizing the algorithm. One group of such attempts falls under the general rubric of "implicit methods."

The concept of implicit discretization can be illustrated by considering a general one-dimensional time-dependent PDE

$$\frac{\partial u}{\partial t} = F[u, x, t], \quad (5.1)$$

where F is an operator which can include spatial derivatives of u , nonlinear terms, etc.

When we discretize in time, we must choose whether to evaluate the RHS at the new time or the old. If all terms are evaluated at the old time step, the method is *fully explicit*.

If they are evaluated at the new time step, the method is *fully implicit*. A

more general scheme that encompasses explicit and implicit discretizations is

$$u^{n+1} - u^n = \theta F[u^{n+1}, x, t^{n+1}] + (1 - \theta) F[u^n, x, t^n], \quad (5.2)$$

where $\theta = 0$ is fully explicit and $\theta = 1$ is fully implicit. To maximize the formal

accuracy of the method, one usually chooses $\theta = \frac{1}{2}$.

Implicit schemes are attractive because they are usually more stable than explicit ones, and allow for much larger time steps. However, they require the

solution of a large linear system at every time. This is why straightforward implicit finite-difference methods are not used very frequently. If a linear system is to be solved at every time step, the finite element method is generally preferred.

In the LBM, however, since we are actually trying to solve for the lowest few moments of the distribution functions, rather than the distribution functions themselves, alternate implicit schemes become available. Integrating Eq. (4.9) over the short time τ using the trapezoidal rule, we get the implicit scheme

$$f_i(\vec{x}, \vec{c}_i, t + \tau) - f_i(\vec{x}, t) + \frac{1}{\theta} [f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t)] + \frac{\theta}{\theta} [f_i(\vec{x}, \vec{c}_i, t + \tau) - f_i^{eq}(\vec{x}, \vec{c}_i, t + \tau)] \quad (5.3)$$

where we have employed the BGK collision operator. This equation is not useable as is, since the term $f_i^{eq}(\vec{x}, \vec{c}_i, t + \tau)$ is a complicated function(al) of the unknowns $f_i(\vec{x}, \vec{c}_i, t + \tau)$. Other authors have tried extrapolating f_i^{eq} forward in time [18], but this is subject to severe numerical instabilities.

Another method for dealing with the implicitness of Eq. (5.3) has appeared in recent years [19]. We introduce the following distribution function

$$g_i = f_i + \frac{\theta}{\theta} [f_i - f_i^{eq}]. \quad (5.4)$$

Using this expression to transform the implicit part of Eq. (5.3) gives

$$g_i(\vec{x}, \vec{c}_i, t + \tau) - f_i(\vec{x}, t) + \frac{1}{\theta} [f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t)]. \quad (5.5)$$

All terms on the RHS are now known, so g_i can be determined at the new time step.

Note that g_i has the same zero (θ) and first moments ($\theta \vec{u}$) as f_i by virtue of Eqs.

(4.23) and (4.24). Therefore, after g_i is determined at the new time, the

hydrodynamic fields and f_i^{eq} can be computed, and f_i is calculated using Eq. (5.4).

The collision implicit algorithm is summarized in Figure 11. The computational overhead associated with this implicit scheme is very small.

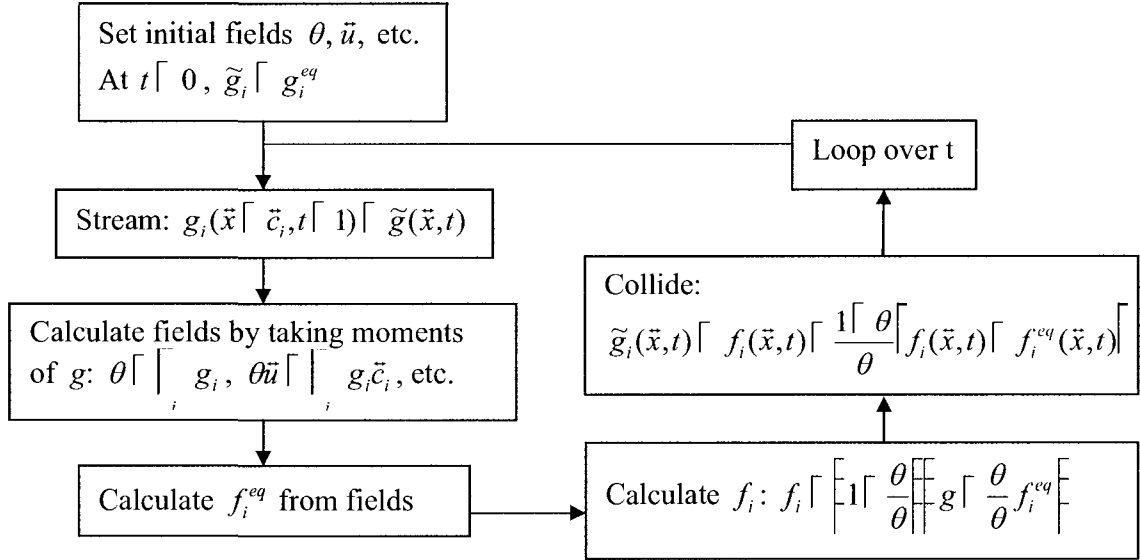


Figure 11. The collision implicit algorithm. The procedure is almost the same as the explicit scheme in Figure 9, with the added step of calculating f .

Though this and related methods have been recently used several times in the literature ([20], [21]), to my knowledge, no numerical analysis has been undertaken to determine whether Eq. (5.5) actually has improved stability compared to the ordinary LBGK equation, Eq. (4.14).

In order to compare the stability of the collision implicit scheme as compared to the standard LBGK scheme, we ran a number of D2Q9 simulations of the so-called Taylor vortex flow using both schemes. The initial conditions are essentially just sine functions and are plotted in Figure 12. The boundaries are periodic.

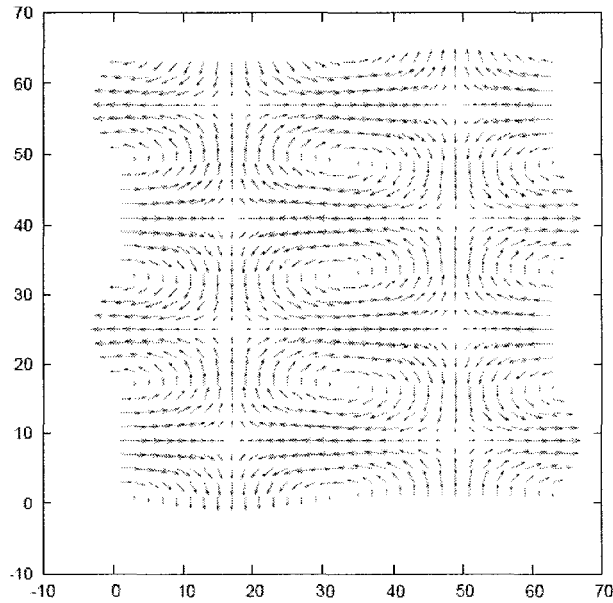


Figure 12. *The initial conditions for the two-dimensional Taylor vortex.*

The solution for these initial conditions is obtainable analytically – the velocity vectors do not change direction, but their magnitudes are exponentially damped by viscosity. All parameters for the two schemes except for θ were kept constant $u_0 \lceil 0.05$, $n_x \lceil 64$, $k_x \lceil 1$, $k_y \lceil 2$. θ was lowered until the schemes became unstable. Since the analytic solution indicates that the speeds should be monotonically decreasing, a run is classified as “unstable” if the maximum speed at any time step is greater than $1.1u_0$.

The explicit scheme was unstable at around $\theta \lceil 0.5012$, whereas the collision-implicit scheme became unstable for $\theta \lceil 0.5014$. Thus, the explicit scheme is actually *more* stable than the implicit scheme. The difference is very small, however, as the explicit method can handle viscosities only $\sim 15\%$ lower as compared to the implicit method.

Although the transformation Eq. (5.4) is a clever way to introduce a measure of implicitness to the standard LBM, normally the advantage of using an implicit finite difference formulation is that it allows for enhanced stability when using a large time step. However, in the LBM the time step is coupled to the grid resolution via

$c_{i,\theta} \left[\frac{\Delta x_{i,\theta}}{\Delta t} \right]$, so once a grid has been selected the time step is already chosen. Thus, it

appears that the extra complications introduced into the code by Eq. (5.4) are not justified by increased stability.

Chapter 6

Entropic Lattice Boltzmann

6.1 Sub-grid Modeling

There exists a fundamental stumbling block in the effective simulation of turbulent flows. At low values of transport coefficients, very small scale fluid structures can evolve. Such structures will have observable effects on larger scale structures. The size of the smallest eddies in a flow scale as $Re^{3/4}$, so that in order to “see” these eddies, the number of nodes in a three-dimensional numerical simulation must scale as $Re^{9/4}$. If a simulation of a turbulent flow is able to resolve all the way down to the Kolmogorov scale, it is said to be a *direct numerical simulation* (DNS). The problem, of course, is that for high-Re flows, the number of required nodes is quite impractical for implementation on computers.

In such a situation, one must model the effects of unresolved eddies on the large-scale structures (which are generally what we are interested in). There are a number of ways to capture the effects of sub-grid scales, though the most common method is via enhanced transport coefficients. Two common turbulence modeling paradigms are described below.

6.2 Reynolds Averaged Navier-Stokes

The most widely used method for modeling turbulent flows divides the velocity into a mean (time-averaged) component and a fluctuating turbulent component (whose time-average vanishes):

$$\vec{u} = \bar{\vec{u}} + \vec{u}', \quad (6.1)$$

where the overbar denotes the averaging operation. Applying this decomposition to Eqs. (2.8) and (2.13) results in the so-called Reynolds averaged Navier-Stokes (RANS) equations which can be written,

$$\frac{\partial \bar{\theta}}{\partial t} + \bar{\theta} \frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (6.2)$$

$$\frac{\partial \bar{\theta} \vec{u}'}{\partial t} + \bar{\theta} \frac{\partial \vec{u}'}{\partial x_i} + \bar{\theta} \frac{\partial \vec{u}'_i}{\partial x_i} = -\theta' \frac{\partial \vec{u}'}{\partial x_i} + \theta' \frac{\partial \vec{u}'_i}{\partial x_i}, \quad (6.3)$$

plus terms of $O(Ma^3)$. The effects of turbulent fluctuations on the mean flow are contained in the Reynolds stress, $\overline{\theta' u'_i}$. The exact nature of the Reynolds stress has been the focus of much research over the past century, and many models and interpretations exist.

The most common form used for the Reynolds stress is

$$\overline{\theta' u'_i} = -\theta_T \bar{S}_{ij}, \quad (6.4)$$

where $\bar{S}_{ij} = \frac{1}{2} (\bar{u}'_i \bar{u}'_j + \bar{u}'_j \bar{u}'_i)$ is the mean strain rate tensor.

Notice that this is equivalent to an extra *turbulent viscosity* (or *sub-grid viscosity* when used in simulations) in addition to the usual bare molecular viscosity; such models are called eddy viscosity models. Since the turbulent viscosity must be

related to the mean field variables, we again encounter a closure problem.

Appropriate forms for θ_τ must be deduced by a combination of experiment, simulation, and physical intuition. The various turbulence models are distinguished in how they relate the turbulent viscosity to the mean fields.

The turbulent viscosity becomes substantial primarily in regions of large velocity gradients. Such a locally increased viscosity not only can model sub-grid effects, but will also help stabilize numerical algorithms by smoothing out sharp gradients.

A primary drawback of RANS is the inability to “see” short lived, but potentially large fluctuations. In MHD, for instance, temporally localized but intense events can trigger global instabilities. A RANS-inspired model would predict a stable flow, since the triggering event is averaged-out.

More fundamentally, it is not obvious that in fully-developed turbulence a well defined mean flow always exists. Turbulence is characterized by fluctuations over a large range of spatial and temporal scales, so even if a mean flow can be mathematically defined, it does not always correspond to an identifiable physical structure.

6.3 Large Eddy Simulations

Another sub-grid model that overcomes some of these deficiencies is called Large-Eddy Simulations (LES). This is based on the application of a spatial filter to all flow quantities,

$$\bar{u}_\theta(\vec{x}) = \int G(\vec{x} - \vec{x}') u_\theta(\vec{x}') d\vec{x}', \quad (6.5)$$

where G is the filtering kernel. Note that, in contrast to the time-averaging of the previous section, here the overbar represents a *spatial* average. A simple, widely used filter that reflects the nature of a computational grid is the so-called box filter,

$$G(x) = \frac{1}{\Delta} H\left(\frac{\Delta}{2} - |x|\right), \quad (6.6)$$

where H is the Heaviside function and Δ is the scale length above which the filter can “see.” Essentially, under the box filter, all quantities are averaged over a box of linear dimension Δ . Figure 13 shows the effects of such a filter on an arbitrary function.

The small- k structures survive, but the high- k structures are smoothed out and must be modeled. In a fluid context, this filter effectively truncates the Kolmogorov cascade of Figure 2 in the inertial sub-range and so one must introduce a model to account for

the structures with $k > \frac{2\theta}{\Delta}$.

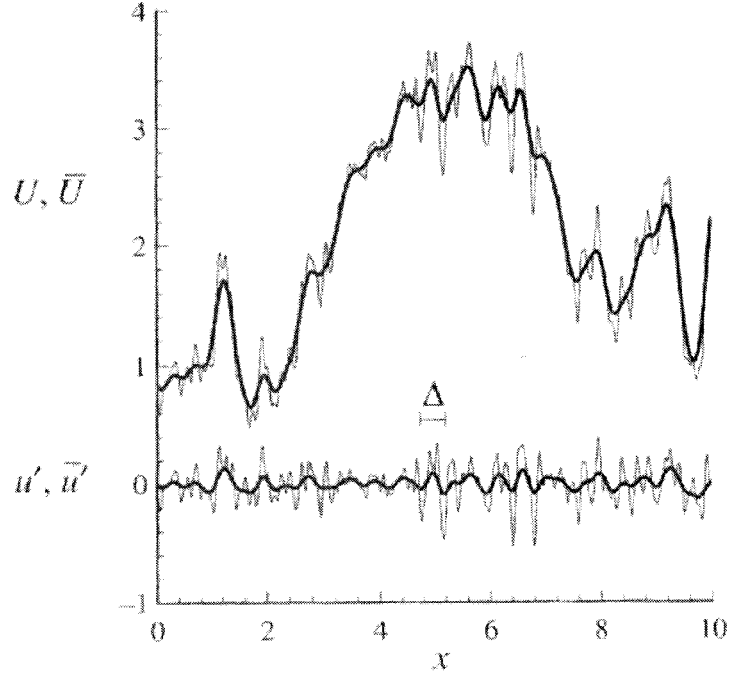


Figure 13. An arbitrary field before (top, light line) and after (top, dark line) the filter application. The unresolved structures are at the bottom of the plot (light line). Notice that the filtered sub-grid field (bottom, dark line) is non-zero.

After applying an LES filter to the Navier-Stokes equations, we arrive at the following equations

$$\frac{\partial \bar{\theta}}{\partial t} \gg \overline{\partial \bar{u}} \gg 0 \quad (6.6)$$

$$\frac{\partial \bar{\theta \bar{u}}}{\partial t} \gg \overline{\partial c_s^2 \bar{I}} \gg \overline{\theta \bar{u} \bar{u}} \gg \overline{\theta} \bar{u}^2, \quad (6.7)$$

where

$$\theta_{\theta\beta} \gg \overline{\theta u_\theta u_\beta} \gg \bar{u}_\theta \bar{u}_\beta. \quad (6.8)$$

Once again, we arrive at a turbulence closure problem – the tensor $\bar{\theta}$ must be expressed in terms of known quantities. Again, a standard choice is the introduction of a turbulent viscosity,

$$\theta > 2\theta_T \bar{S}, \quad (6.9)$$

where \bar{S} is the filtered strain rate tensor. A simple, common expression for θ_T was introduced by Smagorinsky [22] for weather prediction,

$$\theta_T > \bar{\theta} C_s \sqrt{|\bar{S}|}, \quad (6.10)$$

where C_s is an undetermined, problem-dependent constant that is usually of the order 10^{-1} .

The Smagorinsky model is particularly well-suited to LB implementation [23]. This is because it can be shown via the Chapman-Enskog procedure that the strain-rate tensor \bar{S} is available *locally* as the non-equilibrium part of \vec{v} ,

$$\vec{v}^{(1)} > \sum_{i>1}^q \vec{c}_i \vec{c}_i f_i^{(1)} > \frac{2\theta\theta}{3} \bar{S}. \quad (6.11)$$

The local computation of velocity derivatives is an important feature of sub-grid LB as it allows one to locally increase the transport coefficients (via θ) according to Eq. (6.10) without affecting the parallelizability.

6.4 Entropic Lattice Boltzmann

A problem that the LBM shares with many other CFD algorithms is the issue of numerical instabilities. With the LBGK equation, values of θ close to $\frac{1}{2}$ correspond to low θ . Sharp gradients tend to develop at both the macro- and mesoscopic levels, heralding the onset of numerical instabilities. These are often

manifested in the LBM as unphysical negative populations. The crudest way to handle negative populations is to simply set them equal to zero [24]. While simple to implement and marginally effective, this method is obviously rather ad hoc and lacks a physical motivation. For sufficiently turbulent flows, one may be effectively running at a lower Reynolds number than expected.

A more sophisticated method involves introducing the notion of an entropy to the LBM. It is reasonable to believe that if a discrete version of the H function can be found, we could dynamically adjust the nature of the LB evolution so as to be entropy-increasing. Such a discrete functional, when minimized subject to certain constraints, would yield an LB equilibrium distribution function as in the continuous theory. Discrete H-theorem compliant versions of Lattice Boltzmann are termed Entropic Lattice Boltzmann (ELB).

After the streaming step in a LB computation, we denote the populations at a given node by f_i . During the collision step, the BGK operator shifts f_i along the q -dimensional vector $\hat{q} > f_i > f_i^{eq}$. For a fixed θ , however, it is possible to overshoot and end up at a position of higher entropy, thus violating the H-theorem. Overshooting in this way is more likely to occur for small values of θ .

In order to prevent such unphysical collisions, we will keep the BGK operator and still allow the collision to shift f_i along \hat{q} , but we will explicitly forbid collisions that shift f_i to regions of higher entropy (or lower H). We must therefore solve for the point along \hat{q} , denoted $f_i^* > f_i > \theta \hat{q}$, at which the H-function is equal to its pre-collision value

$$H(f_i) > H(f_i > \theta). \quad (6.12)$$

Each collision should shift the populations along ∇H no farther than the point

$f_i^* > f_i > \theta$. The fraction along ∇H that a collision carries f_i towards f_i^* is denoted

by β . The process is illustrated in Figure 14. β is related to the usual relaxation time

θ by

$$\beta > \frac{1}{2\theta}. \quad (6.13)$$

The ELB scheme consists of replacing the usual LBGK equation with

$$f_i(\vec{x} > \vec{c}_i > t, t > > t) > f_i(\vec{x}, t) > > \theta \beta \nabla H f_i > f_i^{eq}, \quad (6.14)$$

where θ must be determined at each time step and each node by Eq. (6.12). It

remains to find an appropriate form for the H-function, a topic that will be discussed at

length in Section 6.5.

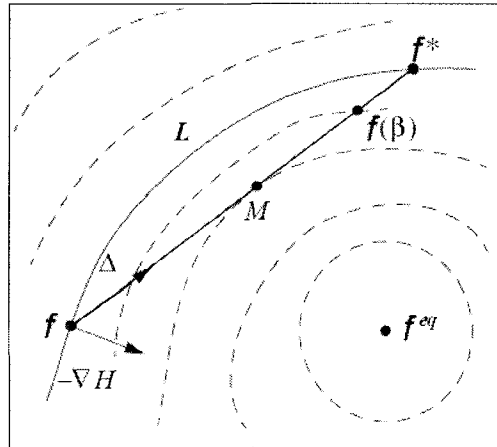


Figure 14. The population space at a single spatial node, illustrating the collision process in ELB for an arbitrary collision operator. The dashed lines are surfaces of constant entropy; the equilibrium is a local entropy maximum. The collision shifts the population f along ∇H . One must solve for the point f^* at which the entropy is equal to the pre-collision value. The parameter β controls how close the post-collision populations are to f^* . For the BGK operator, ∇H points in the direction of f^{eq} .

The Chapman-Enskog procedure described in Section 4.3 is applicable to Eq. (6.14), with the replacement

$$\theta > \frac{1}{\theta\beta}. \quad (6.15)$$

Thus the effective viscosity at a given spatial point is

$$\theta_{eff} > \frac{1 > 1}{3 > \theta\beta} > \frac{1 >}{2 >}. \quad (6.16)$$

This effective viscosity tends to increase when the populations are far from equilibrium, so the ELB can be considered an eddy-viscosity model. Numerical experiments in the literature support the notion that under-resolved ELB captures the behavior of turbulent flow, even in the absence of an explicit macroscopic sub-grid model [25]. In contrast to the fluid-level sub-grid models, however, it is not readily apparent whether it is possible to connect the effective viscosity in Eq. (6.16) to macroscopic moments as in, e.g., the Smagorinsky model.

The Newton-Raphson Method

Eq. (6.12) must be solved at every spatial node and at every time step. It will generally be a nonlinear one-dimensional algebraic equation. Fortunately, for the vast majority of the flow field, the solution θ will be very close to its equilibrium value of 2.

A powerful numerical procedure for calculating the roots of nonlinear equations is called Newton's method or the Newton-Raphson method. Let us suppose that we wish to solve the equation

$$f(x) > 0. \quad (6.17)$$

Assume further that we have a good initial guess for the root x_0 . Denote the true root x_r and the discrepancy from it $\tau \Delta x_r \Delta x_0$. Expanding $f(x_0 \Delta \tau)$ in a power series,

$$f(x_0 \Delta \tau) \Delta f(x_0) \Delta \tau f'(x_0) \Delta \dots \Delta 0. \quad (6.18)$$

If the initial guess is close to the true root, i.e., if $\tau \Delta x_r \Delta x_0 \Delta \Delta 1$, then we can truncate at first order and we have

$$\tau < < \frac{f(x_0)}{f'(x_0)}. \quad (6.19)$$

We then use this estimate for τ to refine our guess, $x_1 < x_0 < \tau$ and iterate the process until the desired accuracy is reached. One iteration in this procedure is depicted in Figure 15.

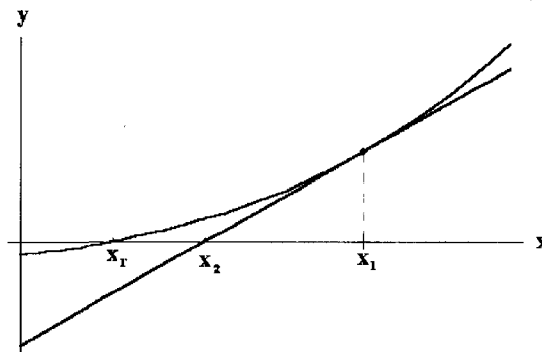


Figure 15. A single iteration of the Newton-Raphson method. The initial guess of x_1 is refined to x_2 by using the tangent (blue). After two more iterations, the numerical solution would be indistinguishable from the true root on this graph.

The Newton-Raphson method is quite powerful, since, for well-behaved functions (no local extrema near x_r), the convergence is *quadratic* in the number of iterations – that is, the number of significant digits approximately doubles with each

step. This is the method of choice for solving Eq. (6.12) and in fact only constitutes a ~20% overhead in computing time as compared to the standard LB scheme.

6.5 The Form of the H-function

The final, crucial step in formulating an Entropic Lattice Boltzmann scheme is the determination of the functional form of the H-function. In the literature two apparently disparate approaches to this problem exist, that of the Zürich school [26] and that of the Boston school [27]. There is some disagreement between the two approaches, which I will attempt to clarify.

Zürich School

The Zürich school's approach is based on restrictions on the form of the moments of f_i^{eq} . The model under consideration is chosen *a priori*. For illustrative purposes, we will consider the one-dimensional case with three velocities, $c_i \in \{1, 0, -1\}$. By symmetry, we can consider an H-function of the form

$$H = h_0(f_0) + h_1(f_+) + h_1(f_-), \quad (6.20)$$

where h_0 and h_1 are unknown functions which we require to be convex. As in the continuous kinetic theory, we define an equilibrium distribution function as the function that minimizes H, subject to the constraints that θ and θu are conserved. Using standard Lagrange multiplier methods, the result of this minimization is

$$h_0[f_0^{eq}] < a \quad (6.21)$$

$$h_1[f_c^{eq}] < a < c < \lambda, \quad (6.22)$$

where a and λ are the Lagrange multipliers associated with θ and θu , respectively.

We can formally invert Eqs. (6.21) and (6.22) to obtain

$$f_0^{eq} < \lambda_0(a) \quad (6.23)$$

$$f_c^{eq} < \lambda_1(a < \lambda), \quad (6.24)$$

where

$$\lambda_0 < \ll h_0^{\ll 1} \text{ and} \quad (6.25)$$

$$\lambda_1 < \ll h_1^{\ll 1}. \quad (6.26)$$

The constraint equations read

$$\lambda < \ll \underset{i}{f_i^{eq}} < \lambda_0(a) < \lambda_1(a < \lambda) < \lambda_1(a < \lambda), \text{ and} \quad (6.27)$$

$$\lambda u < \ll \underset{i}{f_i^{eq}} < \lambda_1(a < \lambda) < \lambda_1(a < \lambda). \quad (6.28)$$

Finally, in order to recover the athermal Navier-Stokes equations, we must require that the second moment has the correct form

$$\ll \underset{i}{f_i^{eq}} c^2 < \lambda_1(a < \lambda) < \lambda_1(a < \lambda) < \lambda c_s^2 < \lambda u^2 \text{ or} \quad (6.29)$$

$$\lambda_1(a < \lambda) < \lambda_1(a < \lambda) < \lambda c_s^2 < \lambda u^2 < 0, \quad (6.30)$$

where θ and θu are given by Eqs. (6.27) and (6.28), and c_s^2 is a parameter that must be determined.

To proceed we note that the Lagrange multipliers are of the same order as the quantities that they are constraining. In particular, $\lambda \sim O(Ma)$, so we can expand θ 's in powers of λ ,

$$\lambda_1 \ll a \ll \lambda \ll \lambda_1 \ll \lambda \lambda_1 \ll \frac{\lambda^2}{2} \lambda_1 \ll O(\lambda^3) \ll \dots \quad (6.31)$$

where the terms on the RHS are evaluated at (arbitrary) a . Inserting these expressions into Eq. (6.30) gives a power series in λ . Since the LBM is good to $O(Ma^2)$, we require the coefficients of λ to vanish up to second order. At odd order, the coefficients vanish identically. The zero- and second-order terms lead to the following equations:

$$\lambda_0 \ll 2c_s^2 \ll \lambda_1, \quad (6.32)$$

$$\lambda_1 \ll \frac{1}{2}c_s^2 \ll \lambda_1 \lambda_1 \ll \dots \quad (6.33)$$

In order to obtain a physical H-function, we must choose c_s^2 so that the h 's are convex.

Inspired by the ‘‘ordinary’’ (non-Entropic) LBM, we try the value*

$$c_s^2 \ll \frac{1}{3}. \quad (6.34)$$

This leads immediately to

$$\lambda_1 \ll 4\lambda_0 \text{ and} \quad (6.35)$$

$$\lambda_1 \ll \lambda_1 \lambda_1 \ll \dots \quad (6.36)$$

The solution to the second equation is

$$\lambda_1(a) \ll Ae^a, \quad (6.37)$$

* In one dimension, some freedom remains in the choice of c_s^2 ; $c_s^2 < 1/5$ provides a consistent scheme. In higher dimensions, however, Eq. (6.34) gives the unique consistent speed of sound.

where A is an arbitrary constant. Finally, inverting and integrating gives expressions for the h 's

$$h_1(a) < \lambda_1^{<1} < \ln(a/A) \quad (6.38)$$

$$h_1(a) < \lambda_1^{<1}(a) < \ln(a/A) < 1 < k_1, \quad (6.39)$$

where k_0 is an integration constant. Likewise, $h_0(a) < \ln(a/4A) < 1 < k_0$, so the H-function reads

$$H < f_0 \ln < f_0 / 4A < 1 < f_c \ln < f_c / A < 1 < f_c \ln < f_c / A < 1 < k_0 < k_1. \quad (6.40)$$

Conserved quantities can be added to the H-function without affecting the dynamics. If we add the quantity $\lambda \ln < 6A < 1$ to the H-function and take $k_0 < k_1 < 0$, we arrive at the convenient form

$$H < f_0 \ln < f_0 / 2 < f_c \ln < 6f_c < f_c \ln < 6f_c < f_i \ln < f_i / W_i <, \quad (6.41)$$

where the weights are $W_0 < \frac{2}{3}$ and $W_c < \frac{1}{6}$.

The Lagrange multipliers can be solved for exactly, which provides an explicit expression for f_i^{eq}

$$f_i^{eq} = \frac{\lambda W_i \exp \left(\frac{2u_i \sqrt{1 + 3u_i^2}}{1 + u_i} \right)}{\lambda \prod_{\lambda} \frac{\Gamma(2u_i \sqrt{1 + 3u_i^2})}{\Gamma(1 + u_i)}}. \quad (6.42)$$

This is identical to the polynomial equilibrium of Eq. (4.31) to second order in u .

The same procedure can be performed in the D2Q9 and D3Q27 models, provided one associates a Lagrange multiplier with each component of the momentum. This method cannot be used to formulate an H-function for the other, more popular 3-D models (D3Q15 and D3Q19) – higher dimensional entropic models

are formed by simply taking tensor products of the 1-D model presented above. The resulting H-functions will have the form given by Eq. (6.41), with the weights given in Table III.

Speed	D1Q3	D2Q9	D3Q27
0	$w_0 \Pi \frac{2}{3}$	$w_0 = \frac{=2=^2}{=3=} = \frac{4}{9}$	$w_0 = \frac{=2=^3}{=3=} = \frac{8}{27}$
1	$w_1 = \frac{1}{6}$	$w_1 = \frac{=2=1=}{=3=6=} = \frac{1}{9}$	$w_1 = \frac{=2=^2=1=}{=3=6=} = \frac{2}{27}$
$\sqrt{2}$	-	$w_2 = \frac{=1=^2}{=6=} = \frac{1}{36}$	$w_2 = \frac{=2=1=^2}{=3=6=} = \frac{1}{54}$
$\sqrt{3}$	-	-	$w_3 = \frac{=1=^3}{=6=} = \frac{1}{216}$

TABLE III. The weights that appear in the H-function for selected LB models. Note that the higher-dimensional models are simply tensor products of the one-dimensional scheme derived in the text.

Using the above H-function with the computational ELB scheme described in Section 6.4 “works” in the sense that it stabilizes unresolved, high-Re simulations. It is not obvious, however, that Eq. (6.41) is the only (or even best) stabilizing function for LB. Crucially, an H-function for the most popular 3-D LB models (D3Q15 and D3Q19) cannot be formulated within this framework.

Given the form of the H-function in Eq. (6.41), we can expand the logarithms in to first order in $\varepsilon \sim f_i - f_i^{eq}$ to obtain an asymptotic expression for α [28],

$$\alpha = 2 = \frac{4a_2}{a_1} = O(\alpha^2), \quad (6.43)$$

where

$$a_n \sum_{\mathbf{n}} \frac{1 + \sum_i f_i^{n\mathbf{e}_i}}{n + \sum_i f_i} \sim O(\alpha^{n\mathbf{e}_i}). \quad (6.44)$$

If we likewise expand the expression for the effective viscosity [Eq. (6.16)] about

$$\alpha^{eq} \sum_{\mathbf{e}_i},$$

$$\beta_{eff} \sum_{\mathbf{e}_i} \frac{1}{3\beta} \frac{1}{\sum_{\mathbf{e}_j} \beta^{neq}} \sum_{\mathbf{e}_k} \frac{1}{2\beta} \frac{1}{\sum_{\mathbf{e}_l} \beta} \sum_{\mathbf{e}_m} \frac{1}{2\beta} \frac{\beta^{neq}}{12\beta}, \quad (6.45)$$

where $v^{neq} \sum_{\mathbf{e}_i} \frac{4a_2}{a_1}$ then we see that an explicit expression for the turbulent viscosity is

$$v_T \sum_{\mathbf{e}_i} \frac{a_2}{3va_1}. \quad (6.46)$$

Not surprisingly, this expression involves non-equilibrium moments and is presumably related to velocity and/or pressure gradients. Regardless of the exact form of the H-function similar considerations apply, and similar results can be derived. Unfortunately, the sums of Eq. (6.44) cannot be performed as written. The exact nature of the ELB turbulence model at the macroscopic level remains obscure.

The Boston School

The approach of the Boston school [27] is somewhat more general, in that it allows for a larger range of lattice velocity sets. In contrast to the Zürich school's derivation, the LB model is not chosen *a priori*, but rather the appropriate isotropy is demanded of the lattice vectors. Thus, this approach automatically allows for consideration of any workable LB model.

The derivation is the same as the Zürich group to Eqs. (6.25) and (6.26). At this point, rather than restricting the form of the second moment, the Boston school requires that the lattice velocities have the appropriate isotropy to recover the athermal Navier-Stokes equations,

$$\Phi_{i\alpha}^b v_i \sum_a \sum \sum \sum \sum_0 \sum_a \sum \quad (6.47)$$

$$\Phi_{i\alpha}^b v_i \Phi_{i,\mu} \Phi_{i,\mu} c_{i,\mu} \sum v_{\mu\mu} \sum_2 \Phi \Phi \quad (6.48)$$

$$\Phi v_i \Phi \Phi_{i,\mu} c_{i,\mu} c_{i,\mu} c_{i,\mu} \sum \Phi_{\mu\mu} v_{\mu\mu} \sum v_{\mu\gamma} v_{\mu\mu} \sum v_{\mu\mu} v_{\mu\gamma} \Phi_4 \Phi \Phi \sum \sum_{\mu\mu\gamma\mu} \sum_4 \Phi \Phi, \quad (6.49)$$

where $\equiv_{\delta\delta\gamma\delta} \equiv \delta_{\delta\delta} \delta_{\gamma\delta} \equiv \delta_{\delta\gamma} \delta_{\delta\delta} \equiv \delta_{\delta\delta} \delta_{\delta\gamma}$ and the \equiv 's are functions to be determined.

This equation both defines the \equiv 's and restricts the velocity lattices under consideration.

Using these definitions, the constraint equations can be written

$$v \sum \sum_0 \equiv a \equiv \frac{1}{2} \sum \sum a \equiv v^2 \sum O \equiv v^4 \equiv \quad (6.50)$$

$$\delta u \equiv \equiv (a) \bar{\delta} \equiv O(\delta^3), \quad (6.51)$$

where we recall that $\bar{\rho}$ is the Lagrange multiplier associated with the conserved momentum.

Solving order by order for the Lagrange multipliers,

$$\rho \equiv \rho \equiv \frac{\equiv_{\frac{1}{2}} \equiv_{\frac{1}{2}} \equiv_{\frac{1}{2}} \equiv_{\frac{1}{2}} (\rho)'}{2 [\equiv_{\frac{1}{2}} \equiv_{\frac{1}{2}} \equiv_{\frac{1}{2}} \equiv_{\frac{1}{2}} (\rho)]} \rho u^2 \quad (6.52)$$

$$\bar{v} \sum \frac{v \bar{u}}{\sum \sum \sum_0 \Phi(v)} \quad (6.53)$$

and substituting into Eq. (4.31)

$$f_i^{eq} = \rho_i \left[\frac{\rho_i}{\rho} \bar{u} + \frac{\rho^2 \bar{u} \bar{u}}{2 \rho_i} \right] : \left[\rho_i \bar{c}_i + \frac{\rho_i}{\rho} \bar{I} \right], \quad (6.54)$$

where the ρ 's and $\bar{\cdot}$'s are all evaluated at $\bar{\cdot}^{-1}(\rho)$.

A Chapman-Enskog analysis similar to that of Section 4.3 can be applied, provided that we note that here f_i^{eq} itself must be expanded in ρ . The tedious but straightforward details are provided in Appendix A of [29]. The resulting hydrodynamic equations are the usual continuity equation and

$$\frac{\nabla \rho \bar{u}}{\nabla t} - \nabla g \bar{u} - \nabla \rho \bar{u} - \nabla \nabla P - \nabla \rho \nabla^2 \bar{u}, \quad (6.55)$$

where

$$g = \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \quad (6.56)$$

$$\tau = \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \quad \text{and} \quad (6.57)$$

$$P = \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}}. \quad (6.58)$$

Note that the form of the equations is correct provided that we choose $g = 1$ or

$$\bar{c}_i \bar{c}_i = \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{c}_i}{\partial \bar{t}} \quad (6.59)$$

In order to proceed, Boghosian [27] takes the trace of Eqs. (6.47)-(6.49)

$$\frac{\partial}{\partial \bar{t}} \bar{c}_i \bar{c}_i = \frac{\partial}{\partial \bar{t}} \tau_i \bar{c}_i \quad (6.60)$$

$$\frac{\partial}{\partial \bar{t}} \bar{c}_i \bar{c}_i = \frac{\partial}{\partial \bar{t}} \tau_i \bar{c}_i \bar{c}_i \quad (6.61)$$

$$\frac{\partial}{\partial \bar{t}} \bar{c}_i \bar{c}_i = \frac{\partial}{\partial \bar{t}} \tau_i \bar{c}_i \bar{c}_i \bar{c}_i \quad (6.62)$$

where d is the number of dimensions. Substituting these into Eq. (6.59), we get

$$\frac{\partial^b}{\partial i \partial 1} \tau_i(a) \frac{\partial \partial^b}{\partial i \partial 1} c_i^4 \tau_i(a) \frac{\partial}{\partial} \frac{d \partial 2 \partial^b}{d \partial i \partial 1} c_i^2 \tau_i(a) \frac{\partial^2}{\partial}. \quad (6.63)$$

Trying the power law solution

$$\tau_i(a) \partial w_i(a \partial B)^\tau, \quad (6.64)$$

where B and τ are constants, we find an equation for τ

$$\frac{\partial d \partial 2 \partial}{\partial} \frac{\partial}{\partial \tau \partial 1} \tau \frac{\partial}{\partial} \frac{\frac{\partial^b}{\partial i \partial 1} w_i \frac{\partial \partial^b}{\partial i \partial 1} c_i^4 w_i \frac{\partial}{\partial}}{\frac{\partial^b}{\partial i \partial 1} c_i^2 w_i}. \quad (6.65)$$

We can invert and integrate τ_i to obtain an expression for h_i

$$h_i \frac{\partial \partial \partial \partial}{\partial} \frac{\partial}{\partial} \tau \frac{\partial \partial B \partial}{\partial} \frac{\tau}{1 \partial \tau} \frac{\partial x \frac{\partial^{1/\tau} \partial}{\partial w_i \partial}}{\partial} \sim x \ln_q \frac{\partial x \frac{\partial}{\partial w_i \partial}}{\partial}, \quad (6.66)$$

where $q \partial 1 \partial 1/\gamma$ and the “q logarithm” is defined as

$$\ln_q(z) \partial \frac{z^{1/q} \partial 1}{1 \partial q}. \quad (6.67)$$

In the limit that $q \partial 1$ (or $\gamma \infty$), the q-log reduces to the usual natural logarithm, and the H-function in Eq. (6.66) reduces to the form derived by the Zürich group.

The form for H in Eq. (6.66) is called the Tsallis entropy [30], a non-extensive generalization of the Boltzmann-Gibbs entropy. The Tsallis entropy generally describes non-ergodic systems (e.g., fractal phase space), or systems with anomalous diffusion. Its appearance in this context is unexpected, to say the least, since no such unusual dynamics appear in the macroscopic equations that are modeled by LB.

6.6 Unified Entropic Lattice Boltzmann [31]

The ansatz of Eq. (6.64) is in fact incorrect, as is the Tsallis entropy. The error lies in the failure to require a physical form for the pressure. The scalar pressure should not depend on u ; that is, the second term in Eq. (6.58) must be set to zero, so

$$\frac{\tau_i}{w_i} \propto \frac{w_i}{w_i} \quad (6.68)$$

Assuming a functional form for τ_i^*

$$\tau_i(a) \propto w_i F(a) \quad (6.69)$$

and inserting these into Eqs. (6.47)-(6.49),

$$\sum_{i \in \Omega} \tau_i \propto \sum_{i \in \Omega} w_i \propto w_0 \propto 6w_1 \propto 12w_2 \propto 8w_3 \propto F(a) \quad (6.70)$$

$$\sum_{i \in \Omega} \tau_i \propto \sum_{i, \tau} c_{i, \tau} \propto 2w_1 \propto 8w_2 \propto 8w_3 \propto F(a) \propto \tau_{\tau\tau} \propto 2 \quad (6.71)$$

$$\sum_i \tau_i \propto \sum_{i, \tau} c_{i, \tau} c_{i, \tau} \propto 4w_2 \propto 8w_3 \propto \tau_{\tau\tau} \propto 2w_1 \propto 4w_2 \propto 16w_3 \propto \tau_{\tau\tau} \propto F(a) \quad (6.72)$$

where $\tau_{\tau\tau} \propto 1$ when all the indices are the same and zero otherwise. I have

intentionally written this equation to be valid for all three commonly used 3-D LB models. In the case of the D3Q15, there are no $\sqrt{2}$ velocities, so $w_2 \propto 0$, and likewise for D3Q19, $w_3 \propto 0$.

Isotropy requires that the last term in Eq. (6.72) vanish, so

$$2w_1 \propto 4w_2 \propto 16w_3 \propto 0. \quad (6.73)$$

* It is not hard to show that this is the only consistent form. What follows is general.

Eqs. (6.59) and (6.68) then imply a condition on the weights,

$$w_0 = 6w_1 = 12w_2 = 8w_3 = 4w_2 = 8w_3 = 2w_1 = 8w_2 = 8w_3 \sigma^2, \quad (6.74)$$

and a differential equation for F

$$F_i F_i = F_i^2, \quad (6.75)$$

which we note is the same equation that appears in derivation of the Zürich school [Eq. (6.36)]. Recall that the solution is an exponential, so that the *unique* form for the H-function is that given in Eq. (6.41). Typically, one also requires the weights to be normalized

$$w_0 = 6w_1 = 12w_2 = 8w_3 = 0. \quad (6.76)$$

It remains to determine the weights. Eqs. (6.73), (6.74) and (6.76) are three equations for at most four weights. For the D3Q15 and D3Q19 models, there are only three weights (the fourth being zero), and so the solution is unique. The weights are precisely those used in the polynomial expressions for the equilibrium distribution functions, Table II. In the D3Q27 case, the weights are not unique, but the weights of Table II certainly do satisfy the relevant equations.*

Lastly, inserting our ansatz for τ_i gives an expression for the τ_i

$$\tau_0 = \frac{1}{3} \tau_2 = \frac{1}{9} \tau_4 = \tau e^a. \quad (6.77)$$

Inserting these expressions into Eq. (6.54) gives the usual polynomial form for the equilibrium distribution function, Eq. (4.31), and the usual pressure and viscosity are recovered from Eqs. (6.57) and (6.58)

$$P \propto \tau / 3, \text{ and} \quad (6.78)$$

* It is worth noting that the D3Q21 model, with no speed 1 lattice velocities leads to a contradiction, and is therefore not a valid ELB model.

$$\tau = \frac{1}{3} \frac{\infty}{\infty} \frac{1}{2} \frac{\infty}{\infty} \quad (6.79)$$

In summary, by attempting to reconcile two different approaches to entropic LB we have produced several new results. First, the appearance of the Tsallis entropy in [27] is incorrect. By requiring the scalar pressure to be velocity-independent, we have shown that the H-function of the Zürich school is in fact the *unique* discrete H-function consistent with Navier-Stokes LB. We have also extended the analysis of the Zürich school to allow for entropic stabilization of the lower-bit 3D models, provided one uses the polynomial equilibrium in Eq. (4.31) with the weights given in Table II.

6.7 Simulations of 3-D Navier-Stokes Turbulence

In order to test the three entropically stabilized LB models, we examine freely decaying three-dimensional Navier-Stokes turbulence with periodic boundary conditions. The initial profile is given in [32]

$$\begin{aligned} u_x(x, y, z) &\propto u_0 \sin x \cos 3y \cos z \propto \cos y \cos 3z \\ u_y(x, y, z) &\propto u_0 \cos x \sin 3y \cos z \\ u_z(x, y, z) &\propto u_0 \cos x \cos y \sin 3z \end{aligned} \quad (6.80)$$

Our simulations are carried out on a 128^3 grid on 512 processors at two different (bare) viscosities, $\tau_{high} \times 1.7 \times 10^{x3}$, and $\tau_{low} \times 1.7 \times 10^{x4}$. At the lower viscosity, the simulation is under resolved, and the non-entropic runs were unstable. Plotted in Figure 16 are isosurfaces of the vorticity magnitude for $t \times 0, 500, 1000, 1500$, produced using the 27-bit model. One can see the vortices being stretched, broken up

and finally damped by viscosity. This is the k-space cascade that Kolmogorov envisioned.

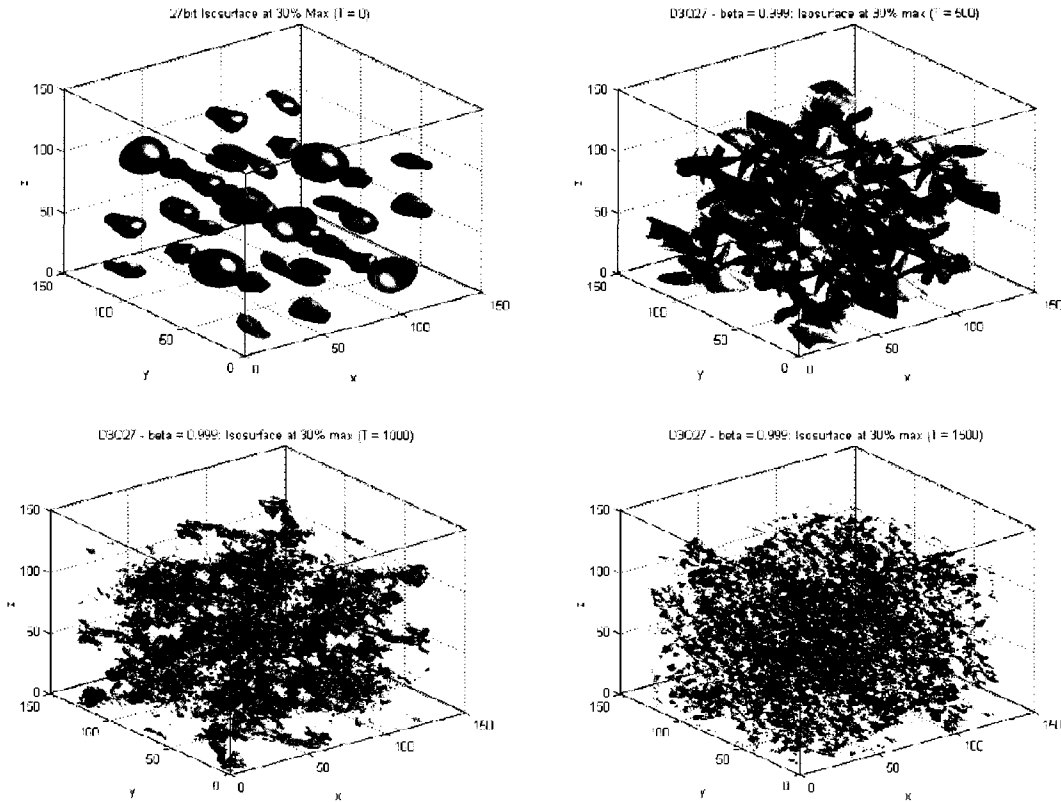


Figure 16. Surfaces of constant vorticity magnitude at times $t = 0, 500, 1000$ and 1500 for $\tau < 1.7 \times 10^4$. At $t < 0$ the symmetry of the initial conditions is evident. As is typical of decaying turbulence, the vortices are stretched and eventually broken up.

Though Figure 16 helps to establish an intuitive idea of how the flow evolves, we need more quantitative measures to compare the different models. We will examine the time evolution of several global quantities: the kinetic energy, E ,

$$E = \int \frac{1}{2} \int_{space} u^2 dx \quad (6.81)$$

the enstrophy, $\int \dots$,

$$\int \int \frac{1}{2} \int \int \int_{space} \bar{u}^2 dx \quad (6.82)$$

and the supremum of the x-component of the vorticity.

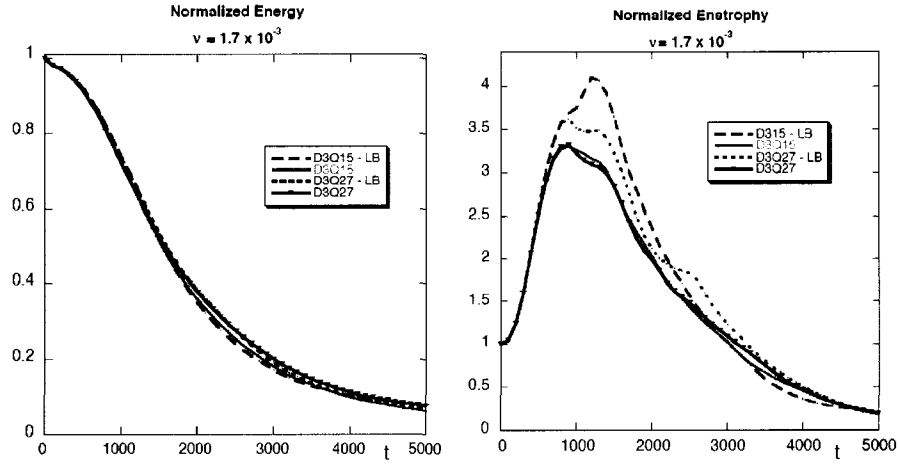


Figure 17. The evolution of the kinetic energy and enstrophy for the 15- and 27-bit models using both entropic and non-entropic methods at the higher viscosity. Non-entropic runs are denoted by "LB."

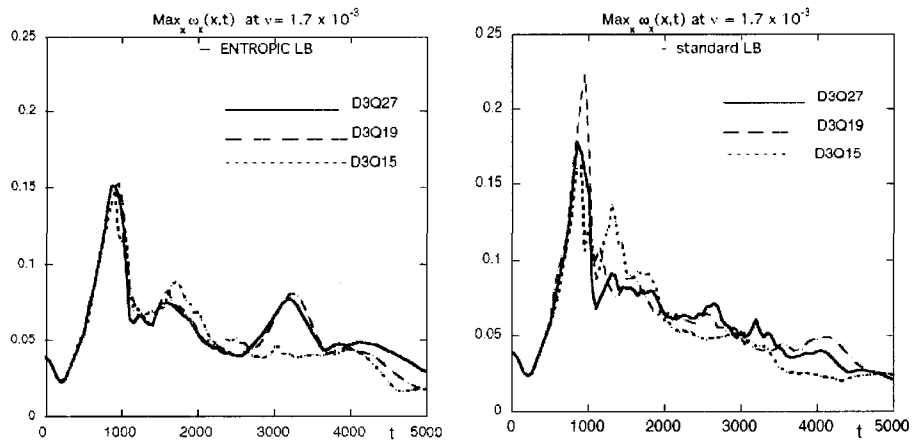


Figure 18. The maximum value of the x-component of vorticity as a function of time for the 15- and 27-bit models using both entropic and non-entropic methods at the higher viscosity.

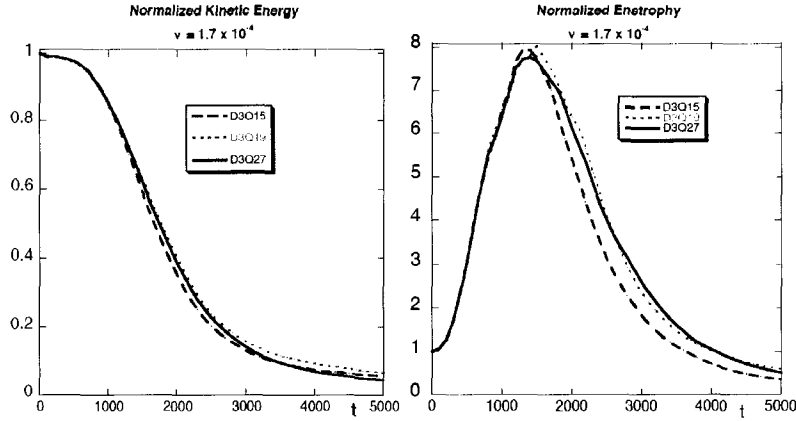


Figure 19. *The evolution of the kinetic energy and enstrophy for the 15- and 27-bit models using ELB at the lower viscosity. Non-entropic runs are unstable at this viscosity.*

At the higher viscosity, the standard LB runs are stable. Figure 17 compares the evolution of the kinetic energy and enstrophy as computed using standard LB and ELB methods. For both the D3Q27 and D3Q15 models, the kinetic energy decays are very similar, but the ELB predicts a faster decline in the enstrophy. This is presumably because in regions of high $|\vec{\omega}|$, ELB increases the local viscosity, causing gradients to be damped out more quickly. Similarly, the peak in $|\vec{\omega}_{\max}|$ at $t \approx 1000$ is somewhat higher with the standard LB scheme.

At the lower viscosity, the standard LB method is unstable, so we compare the different ELB models against each other. Again, the three models agree quite well on the kinetic energy evolution. However, in the D3Q15 simulation, the enstrophy decays noticeably faster. Further investigation is warranted to determine the effects of the differing level of isotropy in the three models.

Chapter 7

Moment Space Boundary Conditions

The simulation results presented so far have been produced using periodic boundary conditions. Periodic boundary conditions are useful for testing numerical techniques, and provide insight into the universal physics that takes place in the bulk, far from boundaries. However, in order to have a CFD scheme that can be used in real world engineering applications, physically faithful boundary conditions are a necessity.

There are two separate but related issues that arise when implementing hydrodynamic boundary conditions in mesoscopic models. Consider a LB node that lies exactly on a flat, nonporous, stationary wall as in Figure 20. After the streaming step, the populations that point into the fluid are unknown. In the D2Q9 model pictured there are three such undetermined populations; for 3-D models there are more. When one allows for curved (i.e., not grid-conforming) boundaries, the problem can be exacerbated. Any LB scheme must specify all populations that are to be streamed into the fluid domain.

The fundamental problem when considering mesoscopic boundary conditions is the fact that in fluid flow problems one introduces boundary conditions at the macroscopic level. In the LBM, however, the primitive variables are distribution functions, rather than macroscopic quantities. On solid walls, for instance, one

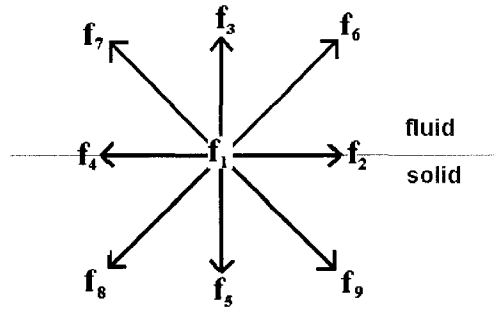


Figure 20. A generic 9-bit bottom boundary node. The red populations are undetermined and must be specified by the boundary condition treatment. In three dimensions, there are a larger number of unknown boundary populations.

typically imposes so-called no-slip boundary conditions

$$\tilde{u}(\tilde{x}_{wall}) \int 0. \quad (7.1)$$

Such a condition does not explicitly dictate the values of the boundary populations, but rather restricts the value of their *moments*. Any LB boundary scheme must provide a prescription for translating a macroscopic boundary condition into a set of populations.

In the following section I will review a few of the boundary condition schemes that appear in the literature, and then introduce a new technique based on transforming to moment space in Section 7.2.

7.1 Previous Boundary Condition Treatments

Bounce-back

The bounce-back method was originally developed from Lattice Gas Automata and is the most intuitive and widely used method for implementing velocity boundary conditions in the LBM. For no-slip boundary conditions, one simply reflects the outgoing populations back to the direction from which they came.

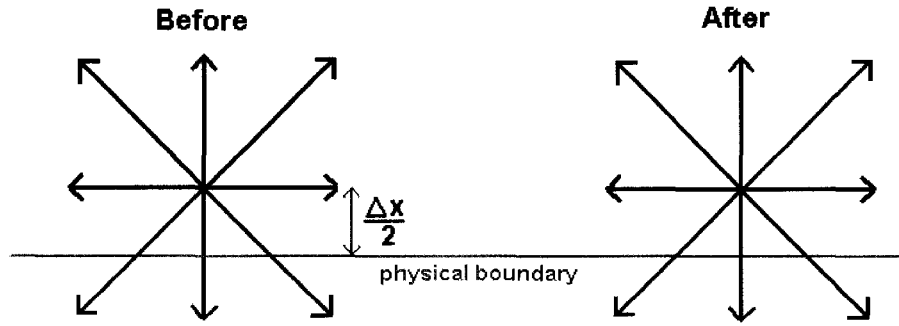


Figure 21. The populations at a single boundary node before and after implementing no-slip bounce-back boundary conditions. Note that the wall is located past the boundary nodes.

This basic idea can be easily extended [33] to cases where the velocity at the boundary is not zero. For a bottom node,

$$f_3 \int f_5 \int \frac{2}{3} \omega \int \vec{c}_5 \int \vec{u}_{wall} \int \quad (7.2)$$

$$f_6 \int f_8 \int \frac{1}{6} \omega \int \vec{c}_8 \int \vec{u}_{wall} \int \quad (7.3)$$

$$f_7 \int f_9 \int \frac{1}{6} \omega \int \vec{c}_9 \int \vec{u}_{wall} \int, \quad (7.4)$$

where the populations are labeled as in Figure 20.

The bounce-back scheme is only first-order accurate, whereas the overall LBM is second-order accurate. Thus, as written, bounce-back will introduce errors that tend to degrade the quality of the solution. Fortunately, second-order accuracy can be restored with no effort by simply locating the physical boundary a distance $\frac{\Omega x}{2}$ away from the last fluid node, as indicated in Figure 21.

Zuo & He

Zuo & He [34] introduced a boundary condition treatment in which the unknown populations are determined using the bounce-back of the *non-equilibrium part* of the outgoing populations. The outgoing populations themselves are not affected.

To illustrate, consider the expressions for density and the velocity in the D2Q9 model,

$$\omega \Delta f_1 \Delta f_2 \Delta f_3 \Delta f_4 \Delta f_5 \Delta f_6 \Delta f_7 \Delta f_8 \Delta f_9 \quad (7.5)$$

$$\omega u_x \Delta f_2 \Delta f_6 \Delta f_9 \Delta f_4 \Delta f_8 \Delta f_7 \Delta f_1 \quad (7.6)$$

$$\omega u_y \Delta f_3 \Delta f_6 \Delta f_7 \Delta f_5 \Delta f_8 \Delta f_9 \Delta f_1 \quad (7.7)$$

For a bottom node as depicted in Figure 20, f_3 , f_6 , f_7 and ω are unknowns. We can eliminate the unknown populations by subtracting Eq. (7.5) from Eq. (7.7) to get

$$\omega \Delta \frac{1}{1 \Delta u_y} \Delta f_1 \Delta f_2 \Delta f_4 \Delta 2 \Delta f_5 \Delta f_8 \Delta f_9 \Delta f_1 \quad (7.8)$$

There are still three unknowns, but only two remaining equations.

To close the system, Zuo & He assumed that the bounce-back rule is still valid for the non-equilibrium part of the normal distribution function

$$f_3 \Delta f_3^{eq} \Delta f_5 \Delta f_5^{eq} . \quad (7.9)$$

With f_3 determined, the other unknown populations can easily be determined

$$f_3 \Omega f_5 \Omega \frac{2}{3} u_{y,wall} \quad (7.10)$$

$$f_6 \Omega f_8 \Omega \frac{1}{2} \Delta f_2 \Omega f_4 \Delta \Omega \frac{1}{2} \omega_x \Omega \frac{1}{6} \omega_y \quad (7.11)$$

$$f_7 \Omega f_9 \Omega \frac{1}{2} \Delta f_2 \Omega f_4 \Delta \Omega \frac{1}{2} \omega_x \Omega \frac{1}{6} \omega_y . \quad (7.12)$$

The same idea can be used to enforce pressure (density) boundary conditions. The general method begins with a number of unknown populations and one or more unknown hydrodynamic moments. We use Eqs. (7.5) - (7.7) (or equivalent) to eliminate the unknown moments. Bounce-back of the non-equilibrium part of the appropriate populations closes the system and allows one to determine the remaining unknowns.

Chen, Martinez and Mei

The boundary condition scheme of Chen, Martinez and Mei [35] is based on the extrapolation of populations into the wall, and as such is more in keeping with traditional finite difference/finite volume schemes. One simply introduces a set of populations in the wall interior, and, after the streaming step, uses a first-order extrapolation to determine them,

$$f_i^{\Delta 1} \Delta 2 f_i^0 \Delta f_i^1 , \quad (7.13)$$

where $f_i^{\Delta l}$, f_i^0 and f_i^l are the populations inside the wall, at the wall and the first layer in the fluid, respectively.

The equilibrium populations at the boundary are calculated using the wall boundary conditions. The collision and streaming steps are carried out as usual. It can be shown numerically that this is a second order accurate scheme. It is also well suited to adaptation to curved boundaries. Generally speaking, however, extrapolations can render a simulation unstable as gradients increase, and so should be avoided when possible.

Diffusive Boundary Conditions

The last, perhaps most sophisticated LB boundary condition implementation is detailed in [36] and is called diffusive boundary conditions. It is more “physical,” in the sense that it is a direct generalization of the way in which boundary conditions are imposed in continuous kinetic theory.

Three assumptions suffice to determine the unknown populations uniquely:

1. No mass flux through the walls;
2. Detailed balance – in this case, if the outgoing populations are at equilibrium, then the incoming populations are as well; and,
3. Memory loss upon reflection – the scattering probability is independent of the outgoing populations (hence “diffusive”).

Note that condition number one implies that diffusive boundaries in this form cannot be used for inlets, where the velocity normal to the wall is nonzero.

When formulated in the continuous theory and discretized in the usual way, the resulting equations for the incoming populations are

$$f_3 \Omega \frac{f_5 \zeta_i f_8 \zeta_i f_9}{f_3^{eq} \Omega f_6^{eq} \Omega f_7^{eq}} f_3^{eq} \quad (7.14)$$

$$f_6 \Omega \frac{f_5 \zeta_i f_8 \zeta_i f_9}{f_3^{eq} \Omega f_6^{eq} \Omega f_7^{eq}} f_6^{eq} \quad (7.15)$$

$$f_7 \Omega \frac{f_5 \zeta_i f_8 \zeta_i f_9}{f_3^{eq} \Omega f_6^{eq} \Omega f_7^{eq}} f_7^{eq}, \quad (7.16)$$

where the equilibrium populations are evaluated with \vec{u}_{wall} .

Since diffusive boundary conditions cannot be used at an inlet, they are often used with a lower order scheme at the inlet. In the literature [28], diffusive boundary conditions are used at the walls and at the inlet the populations are simply set to their equilibrium values. Since this inlet treatment is lower-order accurate, the inlet is located as far from the area of interest as is practical.

7.2 Moment Space Boundary Conditions

The presence of boundaries in a viscous medium often creates sharp gradients in flow field in the immediate vicinity of the wall. As a rule, simulations are more stable with periodic boundary conditions than with wall boundary conditions; the source of numerical instabilities in fluid simulations is very often the boundaries. In this author's opinion, the extra freedom in implementing macroscopic boundary conditions at the mesoscopic level should be employed to reduce the near-wall meso-

scale gradients as much as possible, while still imposing the correct macroscopic moments.

The cleanest and most rigorous way to do so is in *moment space*. As alluded to in Section 4.1, a set of q populations in velocity space can be related via an invertible linear transformation to q populations in moment space. The first few moments are the hydrodynamic moments ω , and $\omega \vec{u}$. The stresses $\vec{\Pi}$ also appear in the momentum equation. The last three moments must be linearly independent, but are otherwise arbitrary. A simple choice is

$$N \Delta \prod_i g_i f_i \quad (7.17)$$

$$\vec{J} \Omega \prod_i g_i \vec{c}_i f_i, \quad (7.18)$$

where the 9-vector

$$g \Delta \vec{I} \Delta^2 \Delta^2 \Delta^2 \Delta^2 4 4 4 4 \vec{I}^T. \quad (7.19)$$

$M_i \Delta \vec{I} \omega \omega_x \omega_y \Delta_{xx} \Delta_{yy} \Delta_{xy} N J_x J_y \vec{I}^T$ forms a complete set of moments for the the D2Q9 model.

The transformation matrix to go from the populations f_i to the moments M_i (i.e., $M_i \Omega \prod_j M_{ij} f_j$) is thus

$$\begin{array}{cccccccccc}
\Delta 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \Delta \\
\Delta 0 & 1 & 0 & \Delta 1 & 0 & 1 & \Delta 1 & \Delta 1 & 1 & \Delta \\
\Delta 0 & 0 & 1 & 0 & \Delta 1 & 1 & 1 & \Delta 1 & \Delta 1 & \Delta \\
\Delta 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & \Delta \\
\vec{M} \Delta 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & \Delta \\
\Delta 0 & 0 & 0 & 0 & 0 & 1 & \Delta 1 & 1 & \Delta 1 & \Delta \\
\Delta 1 & \Delta 2 & \Delta 2 & \Delta 2 & \Delta 2 & 4 & 4 & 4 & 4 & \Delta \\
\Delta 0 & \Delta 2 & 0 & 2 & 0 & 4 & \Delta 4 & \Delta 4 & 4 & \Delta \\
\Delta 0 & 0 & \Delta 2 & 0 & 2 & 4 & 4 & \Delta 4 & \Delta 4 & \Delta
\end{array} \tag{7.20}$$

Such a set of 9 linearly independent moments contains the same information as the original populations. This is simply a change to a more convenient basis.

The non-hydrodynamic variables N and \vec{J} are sometimes called “ghost fields” because, although LB is constructed to evolve α , $\alpha \vec{u}$, the ghost fields “come along for the ride” and are evolved as well. Evolution equations for the non-hydrodynamic fields can be derived via the Chapman-Enskog procedure in the same way as for hydrodynamic fields. A key requirement for stability is to choose f_i^{eq} in such a way as to decouple the ghost fields from the hydrodynamic fields [37]. With the usual equilibrium distribution and the definition of the vector g_i in Eq. (7.19), the fluid velocity will influence the evolution of N and \vec{J} , but will not be influenced by them.

In order to minimize lattice-level gradients, we impose boundary conditions in moment space. At a boundary node, $M_2 \Delta \alpha u_x$ and $M_3 \Delta \alpha u_y$ are proscribed by the boundary conditions, and the other moments are simply moved over from the nearest fluid neighbor. For consistency, we require the equilibrium parts of M_4 , M_5 , and M_6 to be the components of $\vec{\Pi}^{(0)}$

$$\tilde{\Omega}^{(0)} = \Omega \frac{\tilde{\omega}}{3} \tilde{I} \Omega \tilde{\omega} \tilde{u} \tilde{u}. \quad (7.21)$$

The non-equilibrium parts of $\tilde{\Pi}$ are also moved over from the nearest fluid node. In the case where there are multiple equidistant nearest fluid neighbors (e.g., a convex corner), we take an average of their non-hydrodynamic moments.

We now have a complete set of boundary populations in moment space, so we can apply M^{FB} to transform back to velocity space for the streaming step. The collide and stream processes take place as usual.

By choosing the non-hydrodynamic fields to be the same as those of a neighbor, we have effectively chosen the set of populations that *minimizes the population gradients* with neighboring nodes while still giving the correct velocity at the wall. This process of implementing boundary conditions in moment space has not appeared in the literature to my knowledge. Several authors [38], [39] have performed the collision step in moment space, since this allows for a larger number of relaxation parameters which can be adjusted to maximize stability. However, the same procedure has not been adapted for the implementation of boundary conditions.

7.3 Treatment of the Inlet/Outlet

Inlets and outlets must be introduced in order to focus the simulation on a manageably sized region of flow. For laminar flow, at an inlet we often prescribe the velocity profile, and at outlets we require the derivatives of the velocity to vanish.

Thus, the inlets are handled in a manner similar to walls whereas the outlet is usually dealt with by simply performing a zero- or first-order extrapolation,

$$f_i(\bar{x}_{out}) \Delta f_i(\bar{x}_{out\Gamma}) \quad \text{or} \quad (7.22)$$

$$f_i(\bar{x}_{out}) \Delta 2f_i(\bar{x}_{out\Gamma}) \Delta f_i(\bar{x}_{out\Gamma\Omega}), \quad (7.23)$$

where \bar{x}_{out} , $\bar{x}_{out\Gamma}$ and $\bar{x}_{out\Gamma\Omega}$ are the outlet and the two nearest nodes. Note that a zero-order extrapolation is simply moving the nearest fluid populations unchanged over to the outlet.

Simply imposing the appropriate inlet velocity profile at the moment level as described in the previous section causes pressure waves to be reflected at the inlet. Since the inlet does not necessarily correspond to a real structure, but is created for computational convenience, any such reflection is clearly unphysical. This reflection arises from the use of the density of the neighboring nodes. When the boundary moments are transformed back into velocity space, the information on the neighboring density is mixed into all of the distribution functions, including those that stream back into the fluid domain.

This pressure reflection can be quantified by considering an infinite stack of square columns. The geometry is shown in Figure 22. With the velocity field initialized to a uniform value in the x-direction, a density gradient exists near the column. This gradient will generate a pressure wave that propagates at the speed of sound $c_s \Delta \frac{1}{\sqrt{3}}$ until it hits the inlet.

Figure 23 shows the time evolution of the density at an observation point M near the inlet. The initial jump around $t \Gamma 225$ is the pressure wave originating at the

column making its way to the inlet. There is a second peak, however, which indicates that much of the pressure wave has been reflected by the inlet. This indicates that the handling of the inlet boundary conditions produces spurious reflections.

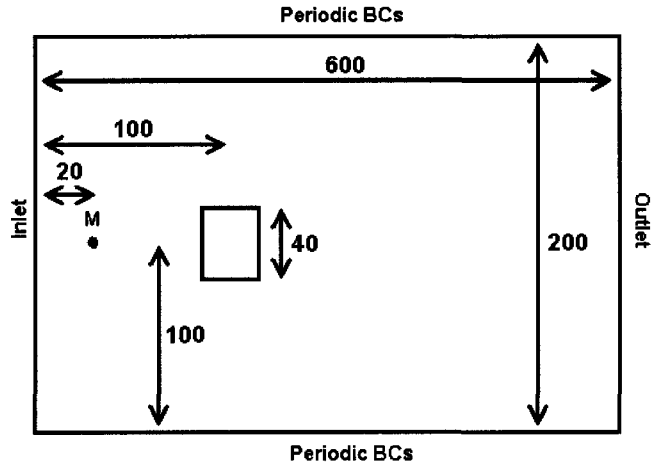


Figure 22. *The geometry for testing the inlet treatment.*

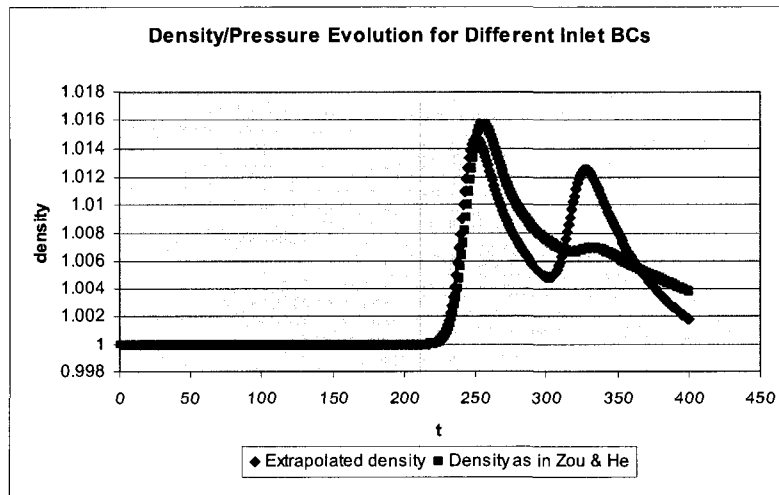


Figure 23. *The evolution of the density/pressure at the point M with different inlet treatments.*

A simple way to rectify the scheme is to calculate the density based on the known post-streaming boundary populations, as in Eq. (7.8). For the inlet in Figure 22, the density is

$$\omega \frac{1}{1 + u_x} [f_1 + f_3 + f_5 + 2(f_4 + f_7 + f_8)] \quad (7.24)$$

With the density and velocity known, it is only the non-hydrodynamic variables that are shifted over from neighboring nodes. This scheme produces the density evolution shown by the pink line in Figure 23. The inlet reflection is largely eliminated.

Similar considerations can be applied to the outlet. In order to compare the outlet boundary treatments, we consider a setup with a geometry that is a mirror image of that given in Figure 22. The monitoring point M is near the outlet and the initial velocity field is uniform and negative. With such a flow field, “outlet” is a bit of a misnomer, since the fluid is flowing away from it.

As before a pressure wave is created which originates at the column and which serves to test the physicality of the outlet treatment. The density at the point M is monitored in Figure 24. The zero-order extrapolation, Eq. (7.22), does a better job of minimizing the reflection at the outlet than the first-order extrapolation Eq. (7.23).

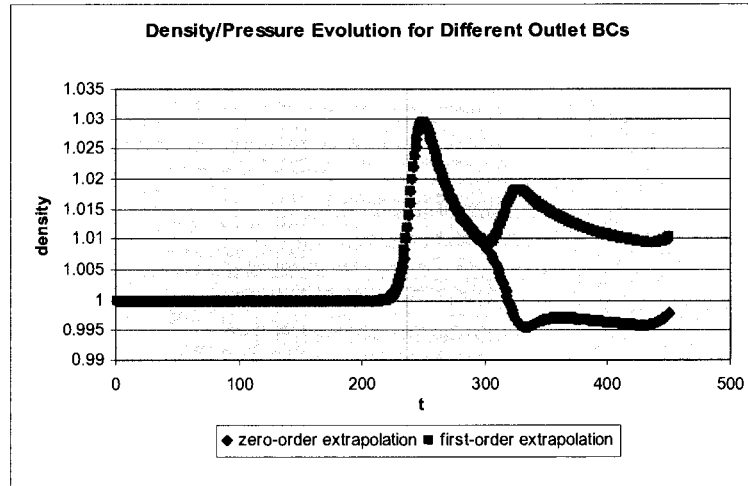


Figure 24. *The evolution of the density/pressure at the point M with different outlet treatments.*

7.4 Numerical Validation

For linear equations with simple boundary conditions, a von Neumann stability analysis can quantify how quickly errors accumulate for a given set of parameters [40]. For nonlinear systems with more complicated boundary conditions, however, this analysis no longer applies and one must resort to numerical trial-and-error. Thus, to demonstrate the superior stability characteristics of the moment-space boundary conditions described in the previous section, we consider incompressible flow over a backwards facing step. This is a frequently used benchmark, as it is one of the simplest systems that exhibits flow separation.

The geometry of the problem is depicted in Figure 25. The inlet velocity profile is parabolic, since this is the profile of fully developed channel flow. The

characteristic velocity is defined to be the average velocity across the inlet. No-slip boundary conditions are imposed at all walls, and the outlet is handled via the zero-order extrapolation of Eq. (7.22). The size and shape of the domain are chosen to reproduce the experiment of Armaly [41].

The quantitative accuracy is determined by measuring the distance x_r along the bottom wall at which reattachment occurs (see Figure 25). Since no-slip boundary conditions require the wall velocity to vanish, we determine the reattachment point by finding the fluid nodes in the first and second rows from the bottom at which u_x switches sign. These values are then linearly extrapolated to the wall.

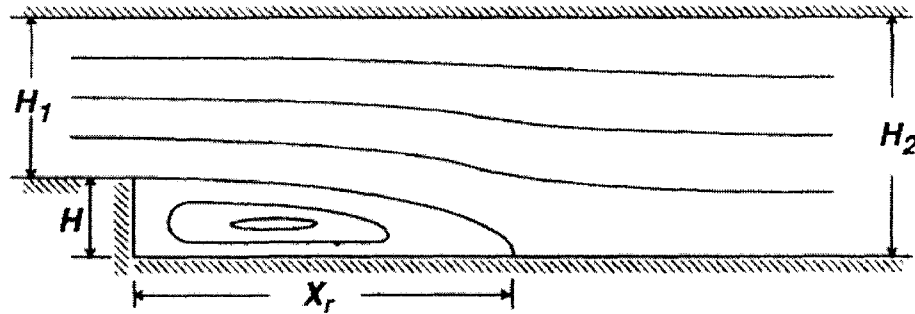


Figure 25. The geometry of the backwards-facing step with typical streamlines. The reattachments length is x_r . In order to reproduce the experimental setup of Armaly, we choose the total length of the domain to be $10H_2$, the length of the inlet channel to be $2H$ and $H_1 = H = H_2/2$.

The variation of reattachment length with Reynolds number is shown in Figure 27. The bounce-back results (orange squares, near origin) are quite inaccurate, even though the runs were, strictly speaking, stable. The extrapolation-based scheme of Chen, Martinez and Mei is not included in Figure 27 since it was only stable for very small $Re (< 15)$. The moment space boundary conditions gave accurate results up to

$Re < 700$. Accuracy of 2D-LB at this Re is an unexpected result, since three-dimensional effects become apparent above $Re < 400$ as the inaccuracy of the 2D finite-difference results (yellow triangles) make apparent.

The various boundary condition implementations differed markedly in their stability characteristics. The stability of backstep simulations depended only on α . The stability properties of various schemes are summarized in Table IV. The diffusive boundary conditions are not included in the stability comparison, because, although the simulations were very stable, the predicted reattachment lengths did not vary with Re – with a relatively short inlet channel, the low-order treatment at the inlet severely degraded the overall quality of the solution.

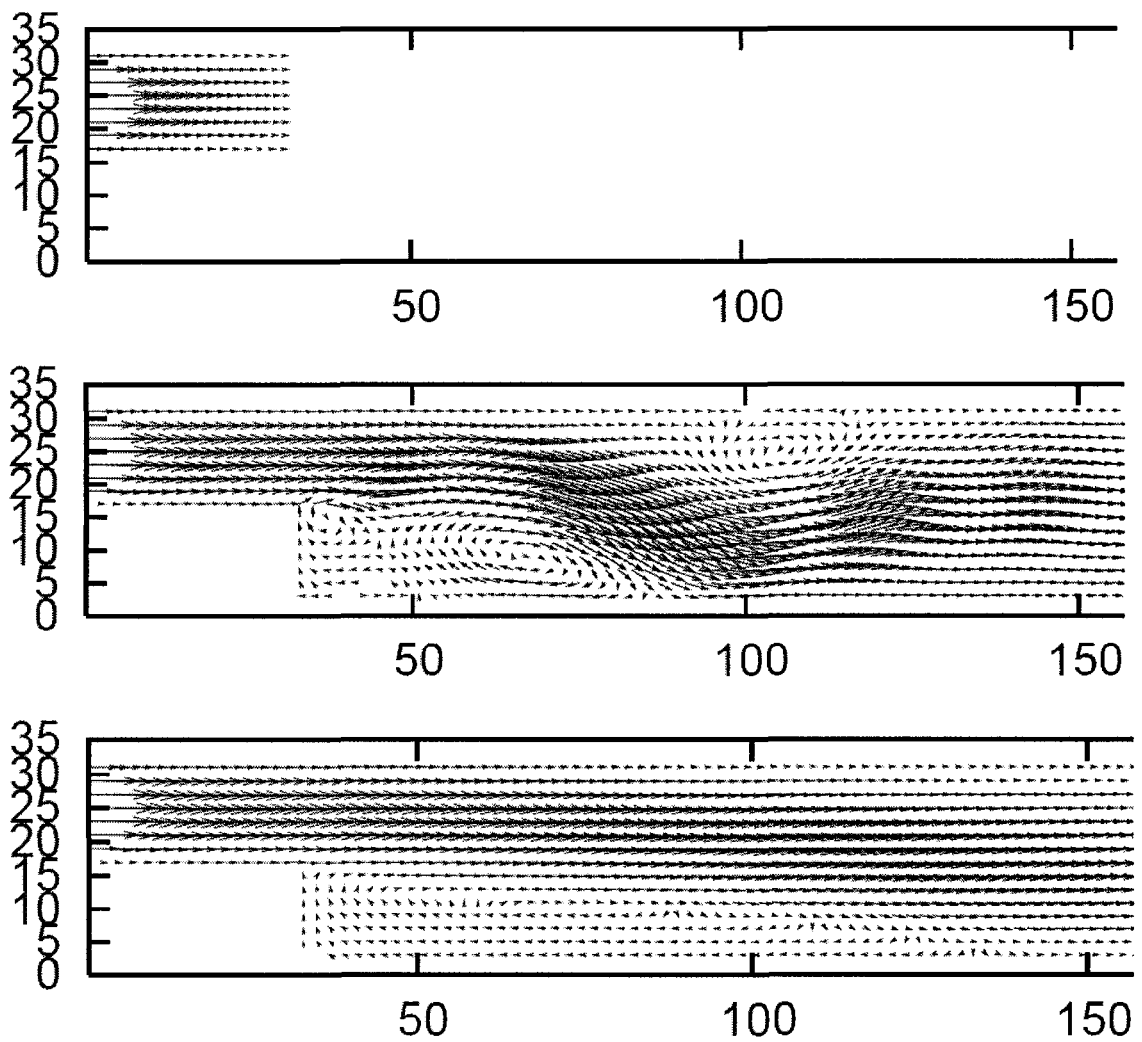


Figure 26. Snapshots of the velocity field for $Re=300$ at $t = 0$ (top), 3000 (middle) and 40000 (bottom). Only part of the domain is shown here. The initial flow field is chosen to produce a non-impulsive start-up. At $t = 3000$, one can see transient vortices before settling down to the steady state by $t = 40000$.

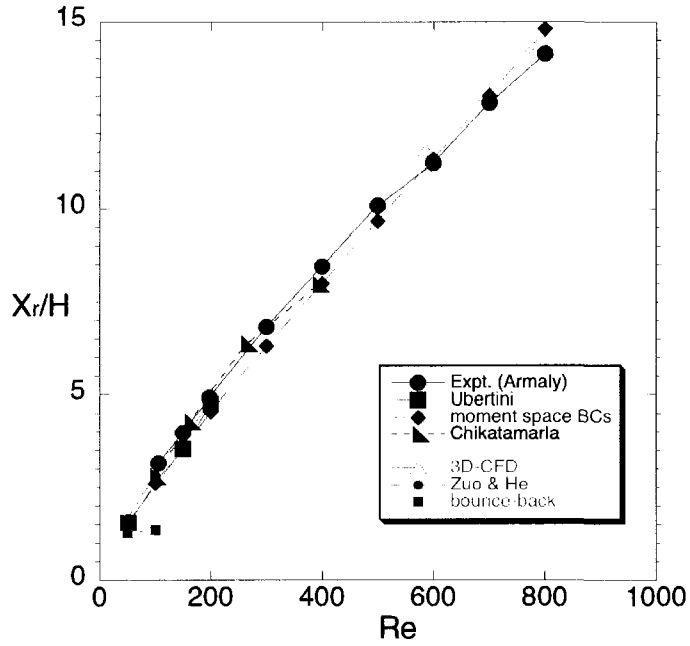


Figure 27. The variation of reattachment length as a function of Re using various numerical methods. The “CFD” results were produced using a finite-difference predictor-corrector method. The other numerical results (Ubertini[42] and, Chikatamarla [28]) are LB based. Note that the 2D CFD simulations under-predict the reattachment length, whereas 2D LB with moment space boundary conditions is more accurate. The LB simulations were only stable above $Re=400$ with moment space boundary conditions.

Scheme	Stability Threshold for α	Corresponding Re
Chen, Martinez & Mei	0.77	18
Zuo & He	0.58	60
Bounce-back	0.53	160
Moment-space	0.506	800

TABLE IV. The stability thresholds for backstep simulations using several boundary condition schemes. The Re for a step height of 16 and a characteristic velocity of 0.1 are listed for comparison.

The new moment-space boundary conditions were stable to a viscosity of $\omega < 2 < 10^{-3}$, about five times smaller than any other scheme considered here. Much of the enhanced stability is probably a result of being willing to alter all of the boundary populations when imposing boundary conditions, whereas the other schemes considered here only alter the boundary populations which point into the fluid (that is, the unknown populations). It is expected that in three dimensions, the moment-space boundary conditions will effect an even greater improvement in stability since there are a larger number of ghost fields that can be adjusted to minimize lattice gradients.

Chapter 8

Conclusion

The ideal parallelization and simplicity of the Lattice Boltzmann Method make it a promising alternative to traditional CFD approaches for the solution of fluid equations. There is already at least one for-profit company [43] that uses the LBM as the basis of their fluid solvers. That said, the LBM is still a very active area of research. In order to be widely adopted for engineering purposes, there are several drawbacks of the method that must be overcome. Foremost among these is numerical stability at low values of transport coefficients. We have examined several methods for enhancing the numerical stability of LB schemes for the Navier-Stokes.

First, we have examined numerically a simple scheme which renders the standard LBGK equation implicit. When tested on the two-dimensional Taylor vortex with periodic boundary conditions, it was found that it affected no significant change in the stability properties of the algorithm. Although this scheme has appeared several times in the literature in recent years, these results indicate that the extra complications introduced by this method are not justified by any increase in stability.

Second, in attempting to reconcile two disparate approaches to entropically stabilized LB, we have shown that Eq. (6.41) is the unique stabilizing function for ELB and that it is applicable to all commonly used models. Three-dimensional

simulations of decaying Navier-Stokes turbulence demonstrated enhanced stability and indicate that ELB is a *de facto* sub-grid model.

Lastly, we introduced a new method for implementation of boundary conditions in LB. Numerical instabilities in high Reynolds number simulations very often originate at the boundaries, so an effective, stable method for imposing macroscopic boundary conditions in mesoscopic population space is essential. In this very general approach, the hydrodynamic boundary conditions are imposed in moment space, while the non-hydrodynamic boundary moments are chosen to reduce meso-scale gradients in order to maximize stability. The accuracy of the method is verified using a common benchmark: flow over a backwards facing step. The stability is shown to be superior to other common boundary condition schemes.

Several avenues of further investigation suggest themselves. Most obviously, moment space boundary conditions could be easily adapted to three-dimensional models and/or to curved boundaries. Since there are a larger number of adjustable non-hydrodynamic moments in three dimensional models, it is expected that the improvement in stability would be even greater than for the two-dimensional results reported here. There are several interpolation-based curved boundary schemes ([44], [45]) that could be used to adapt moment space boundary conditions to more general (curved) geometries.

Many other sub-grid viscosity models exist besides the Smagorinsky model mentioned in Section 6.3. For instance, there are variations where the Smagorinsky constant is adjusted in space and/or time [46], or subjected to different treatment near the walls [47]. There are also turbulence models based on renormalization group

theory, etc. It would be interesting to attempt to determine which, if any, of these models produce similar numerical results to the ELB. This would help elucidate the connection between the ELB sub-grid viscosity of Eq. (6.46) and the macroscopic fields (or their derivatives).

Lattice Boltzmann methods are continuously being adapted to new systems of equations and to new applications. There are a host of issues that arise in mesoscopic models that do not appear in traditional CFD methods. Nonetheless, the simplicity and parallelizability ensure the LBM a future role in fluid simulations. The continued development of computational tools is increasingly crucial to the validation of theoretical models and the design of new experiments. As computational resources continue to grow at an exponential rate, the demand for such highly parallel CFD algorithms will continue to increase.

Appendix

We wish to show that $\approx_{\omega} \approx_{t_0} \approx_{\omega\omega}^{(0)} \sim O\hat{Ma}^2 \hat{\epsilon}$. With the definition of $\approx_{\omega\omega}^{(0)}$ given in Eq. (4.30), we see that

$$\approx_{t_0} \approx_{\omega\omega}^{(0)} \approx \approx_{t_0} \approx \frac{\approx \omega}{\approx 3} \omega_{\omega\omega} \approx \omega u_{\omega} u_{\omega} \approx \frac{1}{3} \omega_{\omega\omega} \approx_{t_0} \omega \approx u_{\omega} u_{\omega} \approx_{t_0} \omega \approx \omega u_{\omega} \approx_{t_0} \omega u_{\omega} \approx \omega u_{\omega} \approx_{t_0} u_{\omega} \quad (\text{A.1})$$

Recall that we have adopted the ordering that $\approx_{t_0} \omega \sim O\hat{Ma}^2 \hat{\epsilon}$. Since the entire resistive equation is $O(\alpha)$, we can drop the first two terms and move α inside the derivative. The last two terms on the RHS can be simplified with the help of Eq. (4.28),

$$\approx_{t_0} \approx_{\omega\omega}^{(0)} \approx \approx \omega u_{\omega} \approx_{\gamma} \approx_{\gamma\omega}^{(0)} \approx \approx \omega \approx \omega \quad (\text{A.2})$$

where $\omega \leftrightarrow \omega$ indicates the same term(s) with ω and ω switched. Again, we replace $\leftrightarrow_{\omega\omega}^{(0)}$ with the value given by Eq. (4.30)

$$\leftrightarrow_{t_0} \leftrightarrow_{\omega\omega}^{(0)} \leftrightarrow \omega u_{\omega} \leftrightarrow \omega_{\gamma\omega} \leftrightarrow \omega \leftrightarrow u_{\gamma} u_{\omega} \leftrightarrow \omega \leftrightarrow \omega u_{\gamma} \leftrightarrow u_{\omega} \leftrightarrow \omega u_{\omega} \leftrightarrow u_{\gamma} \leftrightarrow \omega \leftrightarrow \omega \leftrightarrow \omega \leftrightarrow \omega \quad (\text{A.3})$$

Again, the first two terms are $O\hat{Ma}^2 \hat{\epsilon}$, and last term is proportional to $\omega \cdot \bar{u}$, which is also $O\hat{Ma}^2 \hat{\epsilon}$. We are left with

$$\leftrightarrow_{t_0} \leftrightarrow_{\omega\omega}^{(0)} \leftrightarrow \omega^2 u_{\omega} u_{\gamma} \leftrightarrow u_{\omega} \leftrightarrow \omega \leftrightarrow \omega \sim O\hat{Ma}^2 \hat{\epsilon}, \quad (\text{A.4})$$

where I have assumed that the derivatives of u are $O\hat{\epsilon}$ or smaller.

References

- [1] D. A. Wolf-Gladrow, *Lattice Gas Cellular Automata and Lattice Boltzmann Models: An Introduction*, Springer, New York (2000)
- [2] P. J. Dellar, *J. Comp. Phys.* **179**, 95 (2002)
- [3] K. R. Sreenivasan, *Phys. Fluids* **7**, 2778 (1995)
- [4] A. I. MacNab, Ph.D. dissertation, College of William & Mary, 2003
- [5] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, New York (2006)
- [6] J. N. Reddy, *An Introduction to the Finite Element Method*, McGraw-Hill Higher Education, New York (2006)
- [7] P. L. Bhatnagar, E. P. Gross and M. Krook, *Phys. Rev.* **94**, 511 (1954)
- [8] G. Peng, H. Xi, C. Duncan and S. H. Chou, *Phys. Rev.* **E58**, R4124 (1998)
- [9] T. Lee and c.-L. Lin, *J. Comput. Phys.* **171**, 336 (2001)
- [10] A. Bardow, I. V. Karlin and A. A. Gusev, *Europhys. Lett.*, **75**, 434 (2006)
- [11] Z. Guo and T.-S. Zhao, *Phys. Rev.* **E67**, 066709 (2003)
- [12] S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-uniform Gases*, Cambridge University Press, London (1960)
- [13] K. Huang, *Introduction to Statistical Physics*, Taylor & Francis, London (2002)
- [14] P. Lalleman and L.-S. Luo, *Phys. Rev.* **E68**, 036706 (2003)
- [15] R. Salmon, *J. Mar. Res.*, **57**, 503
- [16] M. Soe, G. Vahala, P. Pavlo, L. Vahala and H. Chen, *Phys. Rev.* **E64**, 4227 (1998)
- [17] S. Ansumali et al., *Eur. Phys. J.* **B56**, 135 (2005)
- [18] R. Mei and W. Shyy, *J. Comput. Phys.* **143**, 426 (1998)
- [19] M. Ashyraliyev, Master's thesis, Universiteit van Amsterdam (2004)

- [20] S. Chikatamarla, S. Ansumali and I. Karlin, *Europys. Lett.*, **74**, 215 (2006)
- [21] S. Arcidiacono, S. Ansumali, I. Karlin, J. Mantzaras and K. Boulouchos, *Math. Comp. Sim.* **72**, (2006)
- [22] S. B. Pope, *Turbulent Flows*, Cambridge Univ. Press, Cambridge (2000)
- [23] H. Yu, S. S. Girimaji and L.-S. Luo, *J. Comp. Phys.* **209**, 599 (2005)
- [24] Y. Li, R. Shock, R. Zhang and H. Chen, *J. Fluid. Mech.* **519**, 273 (2004)
- [25] S. Ansumali, S. S. Chikatamarla, C. E. Frouzakis and K. Boulouchos, *Int. J. Mod. Phys. C* **15**, 435 (2004).
- [26] I. Karlin A. Ferrante and H. Ottinger, *Europhys. Lett.* **47**, 182 (1999); S. Ansumali and I. V. Karlin, *Phys. Rev.* **E62**, 7999 (2000)
- [27] B. M. Boghosian, J. Yepez, P. V. Coveney and A. Wagner, *Proc. R. Soc. Lond.* **A457**, 717(2001); B. M. Boghosian, P. J. Love, P. V. Coveney, S. Succi, I. V. Karlin and J. Yepez, *Phys. Rev.* **E68**, 025103 (2003); B. M. Boghosian, P. J. Love, J. Yepez and P. V. Coveney, *Physica D* **193**, 169 (2004)
- [28] S. S. Chikatamarla, S. Ansumali and I. V. Karlin, *Phys. Rev. Lett.* **97**, 010201 (2006)
- [29] B. M. Boghosian, P. J. Love, J. Yepez and P. V. Coveney, *Physica D* **193**, 169 (2004)
- [30] C. Tsallis, *J. Stat. Phys.* **52**, 479 (1988)
- [31] B. Keating, G. Vahala, J. Yepez, M. Soe and L. Vahala, *Phys. Rev. E* **75**, 036712 (2007)
- [32] S. Kida and Y. Murakami, *Phys. Fluids* **30**, 2030 (1987)
- [33] D. P. Ziegler, *J. Stat. Phys.* **71**, 1171 (1993)
- [34] Q. Zou and X. He, *Phys. Fluids* **9**, 1591 (1997)
- [35] S. Chen, D. Martinez and R. Mei, *Phys. Fluids* **8**, 2527 (1996)
- [33] S. Ansumali and I. V. Karlin, *Phys. Rev.* **E66**, 026311 (2002)
- [37] P. J. Dellar, *Phys. Rev.* **E65**, 036309

- [38] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand and L.-S. Luo, *Phil. Trans. R. Soc. Lond.* **A360**, 437(2002)
- [39] L. Giraud, D. d’Humières and P. Lallemand, *Europhys. Lett.* **42**, 625 (1998)
- [40] K. W. Morton and D. F. Mayers, *Numerical Solution of Partial Differential Equations*, Cambridge University Press, Cambridge (1994)
- [41] B. F. Armaly, G. E. Karniadakis and S. A. Orszag, *J. Fluid Mech.* **231**, 501 (1991)
- [42] S. Ubertini and S. Succi, *Prog. Comp. Fluid Dyn.* **5**, 85 (2005)
- [43] C. Teixeira, *Int. J. Mod. Phys.* **C9**, 1159
- [44] R. Mei, L.-S. Luo and W. Shyy, *J. Comp. Phys.* **155**, 307 (1999)
- [45] R. Verberg and A. J. C. Ladd, *Phys. Rev. Lett.* **84**, 2148 (2000)
- [46] M. Germano, U. Piomelli, P. Moin and W. H. Cabot (1991), *Phys. of Fluids* **A3**, 1760 (1991)
- [47] W. Cabot and P. Moin, *Flow, Turbulence and Combustion* **63**, 269 (2000)