Lattice Boltzmann simulation of flow around bluff-bodies

Kevin Tubbs
Louisiana State University and Agricultural and Mechanical College, ktubbs2@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses
Part of the Physical Sciences and Mathematics Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_theses/3890
LATTICE BOLTZMANN SIMULATION OF FLOW AROUND BLUFF-BODIES

A Thesis

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Master of Science

In

The Department of Physics and Astronomy

by

Kevin Tubbs
B.S., Southern University, Baton Rouge, 2001
May 2004
# TABLE OF CONTENTS

ABSTRACT................................................................................................................................. iii

CHAPTER

1 INTRODUCTION.................................................................................................................. 1
  1.1 Motivation ...................................................................................................................... 1
  1.2 Background .................................................................................................................... 2
  1.3 Originality of Work ........................................................................................................ 3

2 NUMERICAL REVIEW......................................................................................................... 4
  2.1 Grid Based Methods ...................................................................................................... 4
  2.2 Discrete Vortex Methods .............................................................................................. 5
  2.3 Lattice Boltzmann Methods ......................................................................................... 5

3 GOVERNING EQUATIONS ................................................................................................. 8
  3.1 Kinetic Description ....................................................................................................... 8
  3.2 Lengths Scales .............................................................................................................. 8
  3.3 Navier Stokes Equations ............................................................................................... 10
  3.4 The Fundamental Boltzmann Equation ...................................................................... 12
  3.5 Discrete Velocity Boltzmann Equation ...................................................................... 14
  3.6 Lattice Boltzmann Equation ....................................................................................... 16
    3.6.1 Viscosity ............................................................................................................... 19
    3.6.2 Grid Generation ..................................................................................................... 20
    3.6.3 Boundary Conditions ............................................................................................ 21

4 NUMERICAL MODEL .......................................................................................................... 23
  4.1 Computational Domain ................................................................................................. 23
  4.2 Lattice Boltzmann Solver ............................................................................................ 24
  4.3 Force Evaluation .......................................................................................................... 26
  4.4 Results ......................................................................................................................... 27
    4.4.1 Single Bluff Body ................................................................................................. 27
    4.4.2 Multiple Bluff Bodies .......................................................................................... 32

5 CONCLUSIONS.................................................................................................................... 36

REFERENCES ........................................................................................................................ 37

APPENDIX A: CHAMPMAN ENSKOG EXPANSION .............................................................. 40

APPENDIX B: LB SAMPLE CODE ..................................................................................... 43

VITA ........................................................................................................................................ 54
ABSTRACT

In this work, results from a 2-D Lattice Boltzmann (LB) solver are presented simulating flow past rectangular square cylinders at low Reynolds numbers (< 250). The LBGK equation is a hyperbolic equation that approximates the Navier Stokes equations in the nearly incompressible limit. It is a system of 9 one dimensional partial differential Hamiltonian-Jacobian equations, consisting of an advection and diffusive portion. The LB method is an alternative computational fluid dynamics (CFD) method used to numerically predict incompressible viscous flow. The current LB method uses a statistical mechanics formulation to solve the Boltzmann equation.

The LB model captures the nonlinear Navier Stokes advection terms using linear streaming operators. In this thesis, the LB model is classified as an explicit, Lagrangian, finite-hyperbolicity and weakly compressible approximation of the Navier Stokes equations. The momentum flux tensor is captured locally as opposed to a pressure field eliminating the need to solve the Poisson equation. This allows the fluid structure interactions (FSI) behavior to be calculated elegantly at the interface through the mesoscopic momentum transfer between the fluid and structure. At this level, the forces are simultaneously calculated. The LB equations are discretized both in time and phase space using a standard D2Q9 lattice model. Validation tests for flow around single square cylinders at different aspect ratio at low Reynolds numbers are presented. Good agreement with other investigators is achieved. Flow past multiple bluff bodies (representing building in a city) is also presented. The vortex shedding simulations presented provide preliminary indications in terms of St that the LB method can be used to simulate high Re flow.
CHAPTER 1
INTRODUCTION

1.1 Motivation

Fluid flow around bluff bodies is wide area of study with important applications to different areas of science and engineering. These flows range from laminar to turbulent regimes. A measure of laminar or turbulent flow is found through the Reynolds number (defined as \( \text{Re} = \frac{Ud}{v} \), where \( U \) is the constant inflow velocity, \( d \) is the bluff body diameter and \( v \) is the kinematic viscosity). Fluid bluff-body investigations can is experimental, numerical or theoretical. The majority of investigations are experimental. Experimental investigations are mostly limited to lower \( \text{Re} \) (< 5000) due to model scale restrictions. Recently, numerical investigations of fluid dynamics often described as Computational Fluid Dynamics (CFD) are becoming more popular. Numerical investigations are currently capable of modeling fluids at higher \( \text{Re} \) (< \( 10^6 \)); however, much more research is needed to extend current to higher \( \text{Re} \). Numerical investigations are limited only by the computation cost of solving governing equations. The higher the \( \text{Re} \) the more important smaller length scales become, which increases the computational cost. Kinetic theory presents the governing equations at a smaller length scale which may allow smaller length scales to be models with less computational cost. The aim of this work is to apply an alternative formulation based on kinetic theory and statistical mechanics to numerically investigate fluid dynamics around sharp edge bluff bodies and evaluate its usefulness in CFD. In this work, a fluid bluff body model based on the Lattice Boltzmann (LB) equations is presented. Fluid flow is approximated through a 2-D
viscous nearly incompressible fluid flow solver past single and multiple square cylinders. The solutions presented in this thesis are limited to low Re (<250).

1.2 Background

A bluff body is one in which the flow under normal circumstances separates from a large section of body surface thus creating a massive wake region downstream. We consider bluff bodies of various horizontal length to width ratios with sharp edges, i.e. a square cylinder, exposed to a cross flow with a constant free stream velocity, U. Flow separation, boundary layers and shear layers characterize the flow field bluff body disturbance. The innermost portion of the free shear layers moves more slowly than the outermost portion of the layers which are in contact with free stream, the free shear layers tend to roll up into discrete, swirling vortices. At a Re of approximately 50, the vorticial region downstream of the body emanates into the phenomenon of vortex shedding characterized by an unsteady periodic flow situation in which the separated vortices are shed alternately from the upper and lower side of the body. The shedding of vortices is described with reference to the Strouhal number, defined as \( \text{St}=f_s d/U \), where \( f_s \) is the shedding frequency. The near wake flow unsteadiness gives rise to fluctuating drag and lift forces which can result in body vibration. Cylinder vibration can, (i) increase the vortex strength, (ii) increase the span wise wake correlation and (iii) force the shedding frequency to match the natural frequency of the body (lock-in or synchronization). In addition, we note that above a critical Re of around 250, the flow problem may become three dimensional.

Investigations of vorticial instabilities in wakes represent a widely researched area in fluid dynamics. Bluff body vortex dynamics have been studied for more than a century.
One of the first mathematical treatments of vortex shedding was given by von Karman (1911) and the staggered vortex configuration in the wake of a body is therefore usually referred to as the von Karman vortex street. The majority of investigations have been carried out for the flow around a circular cylinder, e.g., see comprehensive review by Zdravkovich (1997). From an engineering point of view, it is also necessary to study flow around other bluff body shapes, such as sharp-edged rectangular cross-sectional cylinders e.g. Knisley (1990). Structures that typically have rectangular or near rectangular cross sections include architectural features on buildings, the buildings themselves, beams, fences and occasionally stays and supports in internal and external flow geometries. The strongest history is still experimental investigations e.g. Williamson (2004). Due to increasing computer power, numerical solutions and insight into the physics have become attractive. Traditionally, CFD solvers are developed based on a continuum mechanics approach. Alternatively, the governing fluid equations can be derived based on kinetic theory and statistical mechanics. Due to its kinetic origin, this approach has some attractive features computationally and physically that may provide more insight and flexibility into understanding fluid bluff-body interactions.

1.3 Originality of Work

To evaluate whether the numerical model presented in this thesis is useful in CFD, the solutions must be validated against other investigators. Investigations of fluid flow around single bluff bodies have been conducted by many investigators. In this thesis, validations are first presented for single bluff bodies after which new contributions are presented in terms of fluid multiple bluff body solutions. This has interest in different areas of science and engineering.
Numerical prediction of fluid flow involves solutions of nonlinear equations. Viscous incompressible flow of a Newtonian fluid is governed by mass continuity and Navier Stokes momentum equations. This set of partial differential equations can be formulated mathematically and solved numerically in a number of ways. Traditionally, numerical models solve a discretized form of the Navier Stokes from a continuum approach. In this chapter three formulations are discussed. Alternatively, the fluid behavior can be described through the Lattice Boltzmann equation whose formulation originates in kinetic theory.

2.1 Grid Based Methods

The approximate, traditional discrete methods used in computational fluid dynamics are usually either the structured finite-difference methods or the discrete weighted-residual methods such as finite-volume or finite-element methods. These methods have been heavily researched over the past 30 years. However, the process of grid generation is not at the same level of maturity as the approximate solution methods, especially for three-dimensional problems, as discussed recently by Douglas et al., (2002). Furthermore, grid generation even in two-dimensional (2D) can be time consuming, especially when the grids involve bluff-bodies.
2.2 Discrete Vortex Methods

A number of numerical models are available that do not require meshing of the flow domain, for example the Discrete Vortex Method (DVM) e.g. Chorin (1973). Chorin also refined the discrete vortex approach by proposing an operator-splitting method which treated the advective and diffusive flow process using separate numerical schemes. To this end, the DVM has been popular for many decades e.g., Leonard, (1980); Stansby & Dixon (1983); and Lewis (1991). In particular, the method of source panels has been used for studying aerodynamic interactions among various components of an aircraft. 2D simulations have been very successful, however, prediction of lock-in in relation to bluff bodies has not, e.g., Larsen & Walther (1998). This is possibly due to problems with replicating turbulent boundary layers e.g., Hunt (2000). Moreover, three-dimensional (3D) simulations are further troublesome. An extensive review of DVM is given by Sarpkaya (1989). The DVM is based on introducing vorticity as a number of blobs at the body surface and tracing their paths through the flow field.

The lack of meshing in the DVM is one its strongest advantages as it ensures that arbitrary geometries can be analyzed. However, the solution tends to become chaotic e.g. Sarpkaya (1989) which can only be avoided by setting certain parameters to obtain the expected solution. This requires several analyses to be conducted to calibrate the model and leaves questions in describing the true physics of the flow by letting the model reach a solution with knowing anything about that solution.

2.3 Lattice Boltzmann Methods

In the past 10-15 years, there has been considerable research in developing and expanding the Lattice Boltzmann (LB) method for different fluid dynamics problems.
Comprehensive reviews of LB methods and the progress of their development have been given e.g., Chen & Doolen (1998); Yu et al. (2003). The LB method was developed as an improvement of the method of lattice gas automata (LGA). It was first introduced by McNamara & Zanetti (1998). Prior to this theoretical contributions have been given e.g., Qian et al. (1992); Chen et al. (1992); d’Humieres (1992). Although inspired by LGA, LB methods are better understood as a self-contained method for solving the Boltzmann equation. The LB method can be developed from fundamental principles, as shown by He & Lou, 1997. The most popular form of lattice Boltzmann equation is the Lattice-BGK (LBGK) incorporating a single time approximation of the Boltzmann equation first introduced by Bhatnagar et al. (1954).

This LB approach is motivated by studies done in molecular dynamics and physics. Although the scope of this research is limited to incompressible fluid dynamics, LB methods are not limited to just that. Research is being conducted in the application of LB methods to other fields such as FSI and blood flow instabilities in artificial heart-value geometries e.g. Krafczyk et. al (2001), compressible flow at high Mach numbers, e.g. Yu and Zhao (2000) and sound wave propagation in 2D urban environment us the related TLM techniques, e.g. Luthi et al. (1996). Generalized LB methods are also being investigated as a discrete numerical solution method for partial differential equations, e.g. numerical solutions to Schrödinger equation in Quantum Mechanics, e.g. Succi (2001). This study is concerned with the LB method as an alternative treatment for fluid dynamics.

The LBGK scheme is a second order method for solving incompressible flows. LB methods differ from Navier Stokes solvers in various aspects including theoretical
and computational interpretations. Some of the advantages of the LB method include a reduction from second order to first order partial differential equations, simplification of non-linear modeling, Poisson freedom, computational efficiency and accuracy, simple fluid interface boundary conditions and a mathematical framework allowing molecular level modeling. The method also has limitations including accurate velocity pressure boundary conditions and grid resolution which lead to low Re modeling. Efforts are currently underway to extend the LB method to higher Re with local grid refinement, force evaluation, numerical stability and turbulence modeling being areas of main concern. Detailed discussions of these issues have been given e.g., Yu et al., (2003); Nourgaliev et al. (2003).

To date, the LB approach has been successfully used for simulating incompressible 2-D and 3-D flows for Re ranging from 200 to 5000 (e.g., Chen & Doolen, 1998; Schafer & Turek, 1996).
CHAPTER 3

GOVERNING EQUATIONS

3.1 Kinetic Description

Kinetic theory assumes that the fluid is described by a large number of molecular constituents whose motions obey Newtonian mechanics. The objective of the theory is not to know the motion of every individual molecule, but the collective behavior for which one needs a statistical description of the system. The statistical description of a fluid at or near equilibrium is contained in the single-particle distribution function, \( f(\mathbf{r}, \mathbf{e}, t) \), where \( \mathbf{r} \) represents spatial coordinates, \( \mathbf{e} \) represents microscopic velocity of molecules and time \( t \). It is defined such that \( [f(\mathbf{r}, \mathbf{e}, t) \, d^3r \, d^3e] \) is the number of particles in a phase space control element \([d^3r \, d^3e]\). The kinetic theory describes the transport equations describing the time and spatial evolution of the distribution function with different collision processes dictated by the nature of the interactions between molecules, given by

\[
\left( \frac{\partial}{\partial t} + \mathbf{e} \cdot \frac{\partial}{\partial \mathbf{r}} + \mathbf{a} \cdot \frac{\partial}{\partial \mathbf{e}} \right) f(\mathbf{r}, \mathbf{e}, t) = \left( \frac{\partial f}{\partial t} \right)_{\text{col}}
\]

where \( \mathbf{a} \) is the external force acting on the particle. In this case the equations are equivalent to the Navier Stokes equations with relevant transport coefficients such as shear and bulk viscosity. The transport equation used is the Boltzmann equation, described in section 3.3.

3.2 Length Scales

The assumption of a fluid at or near equilibrium is fundamental to a kinetic description. Collisions at a molecular level drive the fluid molecules towards a global equilibrium of speed \( U \) and temperature \( T \). The process happens at three different dynamical stages corresponding to three length scales: microscopic, mesoscopic and
macroscopic. Each length scale has a corresponding set of governing equations and time scales. Qualitatively, the approach to equilibrium is controlled by the time scales $\Delta t_c$, $\Delta t_\mu$ and $t_f$, e.g. Bogoliubov (1962). $\Delta t_c$ is defined as the duration of a collisional event and is proportional to ratio of the effective diameter of the particle, $s$, to the velocity of the particle, $e$, (i.e. $\Delta t_c \sim s/e$). This is considered the atomistic or many-body regime where each particle is governed by Newtonian dynamics. $\Delta t_\mu$ is defined as the mean flight-time between two collisions and is proportional to ratio of the mean free path, $l_\mu$, of the particle to the velocity of the particle, (i.e. $\Delta t_\mu \sim l_\mu/e$). This is considered the kinetic regime where a collection of particles are governed by the Boltzmann equation. $t_f$ is defined as the minimum fluid dynamic (convective, diffusive) time scale and is proportional to the ratio of the typical macroscopic scale, $l_M$, to the flow speed, $u$, and the ratio of the typical macroscopic scale squared to the kinematic viscosity, $v$, (i.e. $t_f \sim \min[ l_M/u, l_M^2/v ]$). This is considered the macroscopic regime where an infinitesimal control volume is governed by the Navier Stokes equations. On the time interval, $0 < t < \Delta t_c$, Many body interactions relax many-body distribution functions to the single particle distribution function. On the time interval $\Delta t_c < t < \Delta t_\mu$, the single particle distribution function relaxes to a local equilibrium distribution with smooth space-time dependent flow speed and temperature. On the time interval, $\Delta t_\mu < t < t_f$, the local equilibrium distribution drifts slowly to a global equilibrium distribution with constant macroscopic speed and temperature. These three dynamical stages describe the connections between microscopic and macroscopic length scales using intermediate mesoscopic length scales. In this view, fluid dynamics can be seen as a mean field approximation emerging from a perturbative treatment of the
kinetic equations. Chart 1 summarizes the hierarchy of these length scales increasing from right to left.

Chart 1: Diagram of length scales and corresponding governing equations

![Length Scales Diagram]

- **Atomistic**
  - Newtonian Dynamics
  - 0 < t < Δt_c

- **Kinetic**
  - Boltzmann Equation
  - Δt_c < t < Δt_u

- **Fluid dynamics**
  - Navier Stokes
  - Δt_u < t < t_f

3.3 **Navier Stokes Equations**

In this thesis, incompressible Newtonian fluids and their interaction with sharp edged bluff bodies are of interest. The governing equations can be described by the Navier Stokes equations. The Navier Stokes equation including body forces and turbulence via the Reynolds Stress Tensor is given by

\[
\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = \frac{1}{\rho} \frac{\partial}{\partial x_j} \left\{ -P \delta_{ij} + \mu \left( \frac{\partial U_j}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \rho U_i U_j + F_i \right\}
\]

where \( U \) is the velocity of the flow field, \( u \) is the fluctuating turbulent velocity, \( x \) is the generalized coordinate of the system in the direction of the indices \( i \) and \( j \). The left hand side describes the acceleration and convection of the fluid flow. The dependent variables are the velocity \( (U) \) and the pressure \( (P) \) while the constants are fluid density \((\rho)\) and the
kinematic viscosity (\(\mu\)). The right-hand side represents the mean pressure, viscous and
turbulence effects and the body forces are represented by gravitational force (\(\rho g\)) and
forces due to FSI (\(F_S\)) where \(g\) is gravity. In this thesis, we consider only laminar flow
around fixed bluff bodies. The Reynolds Stress Tensor and body forces are neglected and
(1) reduces to

\[
\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = \frac{1}{\rho} \frac{\partial}{\partial x_j} \left( -P \delta_{ij} + \mu \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right)
\]

Equation (2) can be separated into non-dissipative and dissipative terms through
evaluating the contributions of the momentum flux tensor. The non-dissipative
contribution is given as

\[
\frac{\partial}{\partial x_j} \Pi_{ij} = \frac{\partial}{\partial x_j} \left( U_i U_j + P \delta_{ij} \right)
\]

and the dissipative contribution is given as

\[
\frac{\partial}{\partial x_j} \pi_{ij} = \frac{\partial}{\partial x_j} \left\{ \mu \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right\}
\]

where \(\Pi_{lm}\) is the non-dissipative momentum flux tensor and \(\pi_{lm}\) is the dissipative flux
tensor.

A statistical mechanics approach based on kinetic theory is also capable of
describing a fluid motion governed by the Navier Stokes equations. The present
numerical model is based on kinetic theory which attempts to describe the macroscopic
fluid behavior using the laws of mechanics and probability theory. Provided that the fluid is near a state of equilibrium and the constitutive relations between stress and strain are obeyed, kinetic theory can represent the Navier Stokes equations. In the statistical mechanics approach, the non-dissipative contributions are related to the equilibrium distribution while the dissipative contributions are related to the small departures from equilibrium or non-equilibrium distributions. This connection between the two approaches is given in general as

\[ \Pi_{ij} = m \int f^{(eq)} e_i e_j d\mathbf{e} \]  

(6)

\[ \pi_{ij} = m \int f^{(neq)} e_i e_j d\mathbf{e} . \]  

(7)

The expressions for (6) and (7) are derived in Appendix A.

**3.4 The Fundamental Boltzmann Equation**

The Boltzmann equation was first introduced by Boltzmann (1872). Following the derivation outlined by Nourgaliev et al. (2003), the Boltzmann equation is first presented in dimensional format. The equation is derived by explicitly defining the collision term in equation (1). The Boltzmann equation relates the time evolution and spatial variation of a collection of molecules to a collision operator that describes the interaction of the molecules. Two major assumptions were made in developing the collision operator: (i) only binary collisions are taken into account, and (ii) the velocity of a molecule is uncorrelated with its position. The first assumption is valid if the gas is sufficiently dilute i.e. ideal gas. The second assumption relates to the assumption of molecular chaos in which the collision operator is expressed in terms of the single particle distribution function, \( f \). Without this assumption the collision operator would involve a two particle probability distribution function and in general equation (1) would be replaced by a set of
N coupled equations to account for multi-particle interactions. This set of coupled equation is known as the BBGKY (Bogolyubov, Born, Green, Kirkwood and Yvon) equations. Under these assumptions, Boltzmann expressed the collision term of (1) as

\[
\left( \frac{\partial f}{\partial t} \right)_{coll} = \int d\Theta \int d^3 e \sigma(\Theta) |e-e^{(0)}| (f f' - f^{(0)} f')
\]

(8)

where \( \Theta \) is the scattering angle of the binary collision \( \{e', e^{(0)}\} \rightarrow \{e, e^{(0)}\} \) with fixed velocities \( e, e^{(0)} \); where unprimed quantities \( f, e^{(0)} \) and primed quantities \( e', e'^{(0)} \) denote the velocity and single particle distribution function before and after collision; and \( \sigma(\Theta) \) is the differential cross section of the collision, e.g. Huang (1963).

The collision integral of (8) can be greatly simplified for near equilibrium states by implementing a single time relaxation approximation, (BGK) collision model. The single time relaxation approximation states that during a time interval \( \Delta t_c \) a fraction \( \Delta t_c / \tau = 1/\tau^* \) of the particles in an infinitesimal volume undergoes collisions, drive the single particle probability distribution function to the equilibrium value given by

\[
f^{eq} = \frac{\rho}{(2\pi RT)^D/2} \exp \left[ -\frac{(e-u)^2}{2RT} \right]
\]

(9)

where \( D, R, T, \rho \) and \( u \) are the dimension of space i.e. 2D: \( D=2 \), gas constant, temperature, macroscopic density and velocity, respectively. The BGK collision operator is then given as:

\[
\left( \frac{\partial f}{\partial t} \right)_{coll} = -f - f^{eq} = -f - f^{eq} \frac{\Delta t_c}{\tau^*}
\]

(10)

where \( \tau \) is a relaxation time. The Boltzmann equation with the BGK collision operator is given as
\[
\left( \frac{\partial}{\partial t} + \mathbf{e} \cdot \frac{\partial}{\partial \mathbf{r}} \right) f(\mathbf{r}, \mathbf{e}, t) = -\frac{f - f^{eq}}{\tau}
\]  
(11)

with external body forces, i.e. gravity, electro-magnetic, moving body, etc., are neglected in this model. The fluid flow in this model is completely driven by pressure or velocity boundary conditions.

The mesoscopic quantities of the Boltzmann equation are linked to the macroscopic quantities of fluid dynamics by integration of particle distribution function over momentum space. The macroscopic variables density, velocity and kinetic energy are calculated as the first, second and third moments of the single particle distribution function respectively.

\[
\begin{align*}
\rho &= \int [f] d\mathbf{e}; \quad \rho \mathbf{u} = \int [f \cdot \mathbf{e}] d\mathbf{e}; \quad \rho E = \frac{1}{2} \int [f \cdot (\mathbf{e} - \mathbf{u})^2] \\
\rho &= \int [f^{eq}] d\mathbf{e}; \quad \rho \mathbf{u} = \int [f^{eq} \cdot \mathbf{e}] d\mathbf{e}; \quad \rho E = \frac{1}{2} \int [f^{eq} \cdot (\mathbf{e} - \mathbf{u})^2]
\end{align*}
\]
(12)

with kinetic energy, \(E\), given by

\[
E = N_F / 2 k_B T
\]

where \(N_F\) is the number of degrees of freedom of a particle and \(k_B\) is the Boltzmann constant (\(k_B = 1.3807 \times 10^{-23} \text{ J/K}\)).

### 3.5 Discrete Velocity Boltzmann Equation

The Boltzmann Equation with BGK collision model is discretized in velocity space by introduction a finite set of velocities, \(v_i\), and associated distribution functions, \(f_i(\mathbf{r}, \mathbf{v}_i, t)\). The equation can be non-dimensionalized by the characteristic length scale, \(L\), the reference speed, \(U\), the reference density, \(\rho_r\), and the time between particle collisions, \(\Delta t_c\), giving

\[
\frac{\partial n_i}{\partial t} + \mathbf{c}_i \cdot \nabla n_i = -\frac{1}{\tau_e} \left( n_i - n_i^{eq} \right)
\]
(13)
Equation (13) and all of subsequent equations are presented in non-dimensional variables defined as:

\[ c_i = \frac{\hat{e}_i}{U} \quad \nabla = L\hat{\nabla} \quad t = \frac{\hat{t}U}{L} \]

\[ \tau = \frac{\hat{\tau}}{\Delta t_c} \quad n_i = \frac{\hat{f}_i}{\rho_r} \quad \varepsilon = t_c \frac{U}{L} \]

where \( i \) is the general number of discrete velocities used to approximate the continuous distribution function, e.g. \( i \) can assume values from 0 to infinity. The expansion parameter, \( \varepsilon \), or Knudsen Number can be interpreted as the ratio of collision time to characteristic time. The LB corresponds to a specific discretization of the Boltzmann equation. Equation (13) can be discretized by expansion of particle distribution function in terms of Knudsen Number, choosing \( \Delta t = \Delta \hat{t} U/L \). Then selecting the lattice spacing divided by the time step to equal the lattice velocity: \( c_\alpha = \Delta x^*/\Delta t^* \). The subscript \( \alpha \) denotes the specific discrete velocities of the LB discretization. The set of subscript alphas are a subset of the general set of subscripts \( i \). This results in a Lagrangian formulation of discretized phase space. Choosing \( \Delta t = \Delta t_c \), one has the non-dimensional lattice Boltzmann BGK equation given in non-dimensional lattice units (lu)

\[ n_\alpha (x + c_\alpha \Delta t, t + \Delta t) - n_\alpha (x, t) = -\frac{1}{\tau} (n_\alpha - n_\alpha^{eq}) \quad (14) \]

where the \( \alpha \) represents discrete velocities of the LB discretization of the Boltzmann equation. \( \alpha = 0, 1, \ldots, 8 \) for the 2-D nine speed lattice (D2Q9) and \( \alpha = 0, 1, \ldots, 18 \) for the 3-D fifteen speed lattice (D3Q19). \( \alpha = 0 \) represents the rest particle and none zero \( \alpha \) values correspond to lattice vectors in the direction of nearest neighbors.
The lattice geometry and discrete velocities are shown in Fig. 1. The light center circle is the location of the node and rest particles. Solid and dashed vectors and corresponding dark circles represent particle speeds and locations after one time step. It is noted that the two lattices presented are not the only choices. Lattice geometries are selected based on ability to recover mass, momentum and energy conservation.

In this thesis, the standard LB method presented above is used. The LBGK equation is a hyperbolic equation that approximates the Navier Stokes equations in the nearly incompressible limit. It is a system of 9 one dimensional partial differential Hamiltonian-Jacobian equations, consisting of an advection and diffusive portion. The LB method can be viewed as a special finite difference approximation to solving the discrete velocity Boltzmann equation. Other approaches like FVM have been used, e.g. Peng et al. (1999); Xi et al. (1999).

3.6 Formal Lattice Boltzmann Equation

The formulation presented above is a discretization of the discrete Boltzmann equation which is viewed as an extension of the LGA method. To understand and
improve the LB method for fluid dynamics investigations, a sound theoretical foundation and connection to the continuous Boltzmann equation must be established. He and Lou (1997) demonstrated that the LB equation can be viewed as a specific finite difference (FD) approximation of the continuous Boltzmann equation. This FD approach involves a time discretization coupled to discretization of a 4-D phase space in two dimensions or 6-D phase space in three dimensions. The phase space discretization can be generalized as:

\[
\begin{align*}
  a) & \quad n \rightarrow n_\alpha \\
  b) & \quad \mathbf{c} \rightarrow \mathbf{c}_\alpha \\
  c) & \quad n^\text{eq} \rightarrow n_\alpha^\text{eq} = A_\alpha + B_\alpha c_i u_i + C_\alpha u^2 + D_\alpha c_j c_i u_i u_j
\end{align*}
\]

(15)

where \( \alpha \) is direction of the discrete velocities and \( i \) and \( j = 1, 2, 3 \) are the Cartesian direction of the coordinate system. Macroscopic continuum fluid equations are derived using the multi-scale Chapman Enskog perturbative expansion procedure as derived by Chapman (1970). The discrete equilibrium distribution function in c) is called the Chapman-Enskog expansion. The coefficients \( A_\alpha, B_\alpha, C_\alpha \) and \( D_\alpha \) are such that mass conservation, momentum conservation and viscous stress tensor are recovered during the Chapman Enskog expansion procedure. Through this procedure the equilibrium distribution function is determined to be a constant temperature and small velocity (low Mach number) approximation of the Maxwellian equation (9), as follows

\[
n^\text{eq} \approx \frac{\rho \cdot \exp\left(-\frac{c^2}{2RT}\right)}{(2\pi RT)^{D/2}} \times \left\{ 1 + \frac{(\mathbf{c} \cdot \mathbf{u})}{RT} + \frac{(\mathbf{c} \cdot \mathbf{u})^2}{2(2RT)^2} - \frac{u^2}{2RT} + O(u^3) \right\}
\]

(16)

The phase space discretization establishes the structure of the lattice and the form of the equilibrium distribution function. The discretization must be consistent with the macroscopic variables defined through integration in momentum space, equation (12). In
the derivation by He and Lou (1997), these integral equations have a general form and are approximated by Gaussian quadrature:

$$\int \psi(\mathbf{c}) n^a_{eq}(\mathbf{x}, \mathbf{c}, t) d\mathbf{c} \approx \sum_{\alpha} w_{\alpha} \psi(\mathbf{c}_{\alpha}) n^a_{eq}(\mathbf{x}, \mathbf{c}_{\alpha}, t) \quad (17)$$

Where $\psi(\mathbf{c}) = [1; c_i; (c_i c_j); (c_i c_j c_k); \ldots]$ and $\omega_\alpha$ are polynomials of microscopic velocity, $\mathbf{c}$, and weights of Gaussian quadrature, respectively. Using (17) the LB method can be linked to macroscopic hydrodynamic variables as follows:

$$\rho = \sum_{\alpha} n_{\alpha}; \quad \rho \mathbf{u} = \sum_{\alpha} n_{\alpha} \cdot \mathbf{c}_{\alpha}; \quad \rho E = \frac{1}{2} \sum_{\alpha} n_{\alpha} \cdot (\mathbf{c}_{\alpha} - \mathbf{u})^2$$

$$\rho = \sum_{\alpha} n^a_{eq}; \quad \rho \mathbf{u} = \sum_{\alpha} n^a_{eq} \cdot \mathbf{c}_{\alpha}; \quad \rho E = \frac{1}{2} \sum_{\alpha} n^a_{eq} \cdot (\mathbf{c}_{\alpha} - \mathbf{u})^2 \quad (18)$$

where

$$n_{\alpha}(\mathbf{x}, t) \equiv w_{\alpha} n(\mathbf{x}, \mathbf{c}_{\alpha}, t); \quad n^a_{eq}(\mathbf{x}, t) \equiv w_{\alpha} n^a_{eq}(\mathbf{x}, \mathbf{c}_{\alpha}, t) \quad (19)$$

The selection of the abscissas of the quadrature equation (17) determines the structure or symmetry of the lattice. The details of the procedure to find the required abscissas of the quadrature and corresponding approximations of the Maxwellian are given by He and Lou (1997) for 6-, 7-, and 9-speed lattice models in 2-D and 15- and 27-speed lattice models in 3-D. This is analogous to selecting discrete velocities of the discrete Boltzmann equation. Constraints are imposed on the structure of the lattice based on the Chapman Enskog procedure linking the Boltzmann equation to the Navier Stokes equations. The procedure involves the following moments of the equilibrium distribution function:

**Mass conservation** : $\psi(\mathbf{c}) = [1; c_i; c_j c_k]$

**Momentum conservation** : $\psi(\mathbf{c}) = [1; c_i; c_j; c_j c_k; c_j c_k c_l] \quad (20)$

**Energy conservation** : $\psi(\mathbf{c}) = [1; c_i; c_j; c_j c_k; c_j c_k c_l; c_j c_k c_l c_l]$
The importance of this formulation is that with the chosen abscissas of the Gaussian quadrature equation (17), the moments of the equilibrium distribution function, equation (20) can be treated exactly. The Boltzmann equation in the limits and constraints of the Chapman Enskog procedure are an exact solution to Navier Stokes equations. In this view, the validity of the LB method can rest on the rigorous results of the Boltzmann equation.

3.6.1 Viscosity

The LB method is an approximation of the Navier-Stokes equations. By simulating the dynamics of a fluid at the molecular level, the viscosity is modeled from a molecular point of view. The viscosity of a fluid is a property of the material and in general depends on local density and temperature. In this LB model, temperature is taken to be constant and the viscosity is a function of the density of the fluid. The density of a fluid at a molecular level depends on the mean free path. The mean free path of the fluid can be related to the relaxation time or average time between particle collisions. The viscosity is then a function of the relaxation time whose exact form is determined by deriving the Navier Stokes equations from the LB equation.

In order to derive the Navier Stokes equations from the LB equation, the Chapman Enskog expansion is used (e.g. see Appendix A for full derivation.) The Chapman Enskog expansion corresponds to a multi-scale expansion to first order in space and second order in time. The time and space coordinates are rescaled as

\[ t_1 = \alpha t, \quad t_2 = \alpha^2 t, \quad x_i = \alpha x, \quad \alpha \ll 1 \]  

and the corresponding Taylor expansion of the derivative is

\[
\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2}, \quad \frac{\partial}{\partial x} = \varepsilon \frac{\partial}{\partial x_1}.
\]
The Chapman Enskog expansion assumes that the diffusion time scale, $t_2$, is much slower than the convection time scale $t_1$. The single particle distribution function, $n_{i\alpha}$, is expanded about equilibrium as

$$n_{i\alpha} = n_{i\alpha}^{(0)} + \varepsilon n_{i\alpha}^{(1)} + \varepsilon^2 n_{i\alpha}^{(2)} + O(\varepsilon^3).$$  \hspace{1cm} (23)$$

Applying the Chapman Enskog expansion and taking the incompressible flow limits as described by Chen and Doolen (1998), the traditional continuum mass and momentum are recovered.

Comparing the coefficients of the dissipative flux tensor in the continuum mechanics approach, equation (5), and the statistical mechanics approach, equation (7), the viscosity coefficient is obtained. The expression for viscosity is then given a function of relaxation parameter: $\nu = (\tau - \frac{1}{2})c_s$. In this expression, the relaxation parameter, $\tau$, presents the physical viscosity of the fluid while the factor of $\frac{1}{2}$ is a product of the numerical discretization. This numerical viscosity is an artifact of the lattice and can be considered a propagation viscosity.

### 3.6.2 Grid Generation

The LB method is based on grid bound particles moving along a set of discrete velocities. The computational grid in the LB formulation is a set of spatial points coupled to directional links connecting them forming a lattice. Every position in the lattice can be reached by linear combination translations along the discrete velocities which link them. This allows one to solve equations 2N-dimensional phase space on an N-dimensional computational space. The LB equation (14) is solved on a 2-D computational lattice. The computational grid consists of a 2-D spatial grid of dimensions $n_x \times n_y$ and an $n_v$-dimensional stencil placed at each spatial node. Where $n_x$ is the number of nodes in the $x$
direction and \( n_y \) is the number of nodes in the \( y \) direction and \( n_v \) is the number of discrete velocities of the model. The \( n_v = 9 \) stencil corresponds to the discrete velocities of the D2Q9 velocity model and all possible links between neighboring nodes. Figure 2 shows the regular spatial grid with discrete velocity stencil imposed. In the streaming step a distribution of particles will propagate from each spatial node via one of the eight links or stays at rest on the node. The grid is represented computationally as an \( n_v \times n_x \times n_y \) array.

\[\text{Fig. 2: Diagram of spatial grid and discrete velocity stencil combined to form lattice.}\]

### 3.6.3 Boundary Conditions

Two classes of boundary conditions are frequently encountered in CFD: open boundaries and solid walls. Open boundary conditions include lines or planes of symmetry, periodic cross sections, infinity, and inlet and outlet. On these boundaries, velocity or pressure is usually specified in the macroscopic description of fluid flows. Velocity or pressure conditions can be prescribed using nodal (Dirichlet) or element (Neumann) boundary conditions. A nodal boundary condition prescribes a specific scalar value whereas an element boundary condition prescribes derivatives.
One challenge of the LB method is that the boundary conditions for the distribution functions $n_α$ are not known. One must construct suitable boundary conditions based on macroscopic flow variables. At symmetric and periodic open boundaries, conditions on distribution functions are trivial. Solid boundary conditions can be satisfied approximately by solving for the unknown distribution functions ($n_α$). After the collision step, $n_α(x_f)$ at the fluid node $x_f$ in the fluid region is known for all $α$, but $n_α(x_b)$, the distribution function streaming from the solid node $x_b$ to a fluid node $x_f$ is not known. To complete the streaming step, $n_α(x_b,t)$ is needed because it exactly gives $n_α(x_b,t+δt)$ after streaming. For the no-slip boundary condition, these distribution functions must be chosen such that the macroscopic velocity is set to zero on the boundary. A popular boundary condition is to employ a bounce back scheme, as described by Ziegler (1993). In this scheme, the momentum from post collision particle is bounced back in the opposite direction after the particle hits the wall. Reversing the momentum attaches the particle to the solid surface and sets the particle’s velocity equal to the velocity of the solid wall. For the free slip boundary condition, the distribution functions are chosen such that the tangential motion of the fluid flow is free and no momentum is to be exchanged with the wall along the tangential direction. This is achieved in a similar manner to the bounce back scheme; however, the only the normal components are reversed while the tangential components are allowed to stream freely.
CHAPTER 4

NUMERICAL MODEL

4.1 Computational Domain

The boundary conditions prescribed on the computational domain shown in Figure 3. At the inlet boundary, $\Gamma_1$, a constant velocity profile is prescribed using equilibrium distribution functions applied to the first lattice column. At the top and bottom boundary conditions, $\Gamma_2$ and $\Gamma_3$ respectively, free-slip boundary conditions are applied with free stream velocity. The top and bottom boundaries are located a distance $(Z-d)/2$ above and below the bluff body. At the outlet of the domain, $\Gamma_4$, a constant pressure and zero velocity gradient in the x direction is applied. The outlet is located a distance $Y$ away from the bluff body. No-slip boundary conditions are prescribed at the walls of the bluff-body, $\Gamma_5$. The bluff body is located a distance $X$ from the inlet. The no-slip boundary condition implemented in this thesis, is based on second order accurate boundary condition (half-bounce back scheme) as described by Chen & Doolen, (1998).

Fig. 3: Computational domain and boundary conditions.
4.2 Lattice Boltzmann Solver

The majority of grid based methods presented in section 2.1 are solved via primitive variable solutions which have advantages and disadvantages. An advantage of primitive variable solution procedures is that they can be applied to 3-D flow problems in a straightforward manner. A common feature in these methods is the solution of the Poisson equation for pressure or pressure correction. Solving this equation is the majority of the computational cost and represents the main disadvantage.

Incompressible viscous fluid flow is described mathematically by the Navier Stokes equations consisting of momentum equations coupled to the continuity equation. These equations have four unknowns in 3-D and three unknowns in 2-D, corresponding to the pressure and components of velocity. Solving these equations as one large non-linear system of equations is very expensive to solve. For this reason, most solvers solve pressure and velocity weakly coupled. These solution methods can be categorized into two main types: artificial compressibility and pressure correction methods. The artificial compressibility method was originally proposed by Chorin (1967). It is based on solving a modified form of the continuity equation for compressible flow. At each time step, a Poisson equation is usually solved for pressure to drive the flow solution to a steady state.

There are several algorithms that can be considered pressure correction methods such as the Marker and Cell, (MAC) method proposed by Harlow and Welch (1965) and the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) method first introduced by Patankar and Spalding (1972). In these methods, at each time step, an intermediate velocity is estimated. The estimated velocity is used to solve the Poisson equation to obtain a new corresponding pressure field. Then steps are taken to advance
both the pressure and velocity in time. Different variations of this method have been used to simulate flow past bluff bodies, (e.g. Davis and Moore 1982; Okajima et al. 1993). The SIMPLE algorithm involves solving a Poisson equation for a pressure correction instead of pressure. This improves convergence compared to the projection method because the difference between estimated and final velocities is smaller. Variations of the SIMPLE method have been developed to increase computational efficiency and stability, e.g. SIMPLER, e.g. Patankar and Spalding (1972) and SIMPLEC, e.g. Pierre (1988). These algorithms are employed to obtain a steady state at each time step.

The LB solver in this thesis has the advantage of straightforward 3-D implementation but eliminates the disadvantage of solving the Poisson equation. To solve for \( n_\alpha \) numerically, equation (14) is solved using a basic stream and collide algorithm. The advection part is performed in the streaming part of the algorithm and diffusion in the collision part. Equation (14) is computed in two steps:

collision step:

\[
\tilde{n}_\alpha(x, t + \Delta t) = n_\alpha(x, t) - \frac{1}{\tau} \left[ n_\alpha(x, t) - n_\alpha^{(eq)}(x, t) \right]
\]  \hspace{1cm} (24a)

streaming step:

\[
n_\alpha(x + c_\alpha \Delta t, t + \Delta t) = \tilde{n}_\alpha(x, t + \Delta t)
\]  \hspace{1cm} (24b)

Where \( \sim \) denotes the post collision state of the distribution function. Note the left-hand-side of (24a) and the right-hand-side of (24b) are at time level \( t + \delta t \) as these equations are solved explicitly. The equilibrium distribution for is given as:

\[
n_\alpha^{(eq)} = \rho \omega_\alpha \left[ 1 + \frac{3}{c^2} \mathbf{c}_\alpha \cdot \mathbf{u} + \frac{9}{2c^4} (\mathbf{c}_\alpha \cdot \mathbf{u})^2 - \frac{3}{2c^2} \mathbf{u} \cdot \mathbf{u} \right]
\]  \hspace{1cm} (25)

where \( \omega_\alpha \) is the weighting factor of a Gaussian quadrature given by
\[
\omega_\alpha = \begin{cases} 
\frac{4}{9}, & \alpha = 0, \\
\frac{1}{9}, & \alpha = 1, 2, 3, 4 \\
\frac{1}{36}, & \alpha = 5, 6, 7, 8 
\end{cases} 
\]  

(26)

Macroscopic variables density and momentum in phase space are calculated as the moments of the distribution functions given in equation (18). The pressure is determined from the pseudo equation of state for an ideal gas: \( p = \rho c_s^2 \), where \( c_s \) is the pseudo speed of sound in the lattice model.

### 4.3 Force Evaluation

Estimation of fluid forces on bluff bodies can be handled either by pressure stress integration approach or momentum exchange approach. Comparisons of these methods were performed by Yu et al., (2003) and the momentum exchange approach was found to be more accurate because it avoids extrapolations. For this reason, the momentum exchange method was chosen for force estimation. The total resultant fluid force, \( F \), on a fixed bluff body is obtained as:

\[
F = \sum_{\beta} \sum_{\alpha} n_{\alpha}(\mathbf{x}_b, t) \delta \mathbf{x} / \delta t
\]

(27)

where \( N_d \) is the number of non-zero lattice velocity vectors, the subscript \( \beta \) denotes the lattice direction opposite of the \( \alpha \) direction, \( \alpha = -\beta = 1, 2, \ldots 8 \) in 2-D and 1, 2, \ldots, 18 in 3-D. Equation (4) is evaluated at the midpoint of the fluid lattice nodes at \( \mathbf{x}_f = (\mathbf{x}_b + \mathbf{e}_\beta \delta t, t) \) and the solid lattice nodes at \( \mathbf{x}_b = (\mathbf{x}_f + \mathbf{e}_\alpha \delta t, t) \) giving the fluid solid momentum exchange per unit time. Note the distribution functions used here are in a post collision state. The inner summation describes the momentum exchange between a solid node at \( \mathbf{x}_b \) and all possible neighboring fluid nodes around that solid node. The outer summation includes the force contributed by all boundary nodes \( \mathbf{x}_b \). Equation (4) is applicable in both 2-D and 3-D LB.
models. The estimated force is used to calculate characteristic drag, $C_d$, and lift, $C_l$, coefficients defined as:

$$ C_d = \frac{F_x}{\frac{1}{2} \rho U^2 H} \quad C_l = \frac{F_y}{\frac{1}{2} \rho U^2 d} $$

### 4.4 Results

In the following, solutions are presented for flow past fixed single and multiple bluff bodies on 2-D uniform grids.

#### 4.4.1 Single Bluff Body

First, solutions are presented for flow around a single square cylinder for Re = 100 and 250. The computational domain is $100d \times 30d$ where $d=0.166$ m denotes the cylinder diameter. The cylinder is located at $X=20d$ from the inlet and vertically centered ($Z/2$) in the domain. The outlet is located at $Y=80d$.

Simulations were carried out for three grid resolutions, 500×80, 1000×160, and 2000×320 with 10, 20, and 40 lattice nodes on the diameter respectively. $\Delta x$ and $\Delta t$ in lattice units (lu) are coupled by the uniform lattice spacing of $\Delta x/\Delta t = 1$. The lattice units are related to physical units through the Re. LB parameters are chosen so that the Re in lattice units equal the Re in physical units. These LB parameters translate into physical spatial and time steps given in Table 1.

<table>
<thead>
<tr>
<th>Grid Resolution</th>
<th>$\Delta x$ (m)</th>
<th>$\Delta t$ (s)</th>
<th>$\tau$ (lu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500×80</td>
<td>0.00166</td>
<td>0.0018</td>
<td>0.53</td>
</tr>
<tr>
<td>1000×160</td>
<td>0.00083</td>
<td>0.0009</td>
<td>0.56</td>
</tr>
<tr>
<td>2000×320</td>
<td>0.00042</td>
<td>0.0005</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Grid sensitivity test were carried out. Grids 2 and 3 converged to the same Strouhal number, St. Therefore, the remaining simulations relate to grid 2. St of 0.148
and 0.158 was obtained for Re= 100 and 250 respectively. These LB results fairly agree with the results obtained by other investigators, as shown in Table 2.

<table>
<thead>
<tr>
<th>Reynolds No.</th>
<th>100</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present LB Model</td>
<td>0.148</td>
<td>0.158</td>
</tr>
<tr>
<td>Okajima (1982)</td>
<td>0.141-0.145</td>
<td>0.139-0.143</td>
</tr>
<tr>
<td>Davis et al. (1984)</td>
<td>0.154</td>
<td>0.165</td>
</tr>
<tr>
<td>Sohankar et. al (1998)</td>
<td>0.147</td>
<td>0.154</td>
</tr>
</tbody>
</table>

Davis et al. (1982) implemented a 2D Finite Volume (FV) solver to simulate flow over rectangular cross sections in infinite and confined domains. Davis et al. showed good results for Re below 1000. Franke et al. (1990) implemented a 2D FV third order-accurate scheme to solve unsteady flow past a square cylinder for Re≤300. We note that Franke et al. reported on numerical problems due to sharp edged geometry. Okajima et al. (1992) implemented a 2D Finite Difference (FD) solver for low Re and a Discrete Vortex (DV) method for high Re on flow past rectangular cylinders of varying aspect ratio. Low Re results successfully captured changes in flow pattern. Okajima simulated high Re and found jumps in St. Mahir (2002) implemented a different high order FD solver for flow around square cylinder for Re ≤250 with results similar to experimental results. Sohankar et al. (1999) developed a FV SIMPLE code for flow around sharp edge bodies in 2D and 3D.

In the following, the horizontal length, H, to diameter aspect ratio, (a=H/d), is varied for the cases of a=2,3 and 4 with same computational domain for Re=100 as shown in figure 5. St, lift, C_l and drag, C_d are predicted. Figure 4 shows the time history of drag and lift force coefficients for a=1. The time histories are harmonic with the lift coefficient fluctuation about a mean close to zero as expected. Solutions are listed in Table 3 for different sharp edge bodies. The solutions are compared with Sohankar et al.
(1998) who used a FV SIMPLE Navier Stokes Solver on a non-uniform grid. The non-uniform grids include high grid resolution near the boundary of the bluff body, as steep velocity gradients can occur at high Re. The present LB model is based on uniform grids but captures the similar responses for various aspect ratios. Compared to Sohankar et al. good agreement is found for the St; however, drag coefficients are higher in the present LB model. This over prediction is most likely due to uniform grid resolution, i.e. insufficient grid points in boundary layer.

Table 3: Drag and St of single sharp edged bodies (Re=100).

<table>
<thead>
<tr>
<th>a</th>
<th>Present LB Model</th>
<th>Sohankar (1999)</th>
</tr>
</thead>
<tbody>
<tr>
<td>St 1.0</td>
<td>0.148</td>
<td>0.147</td>
</tr>
<tr>
<td>Cd 1.694</td>
<td></td>
<td>1.444</td>
</tr>
<tr>
<td>St 2.0</td>
<td>0.135</td>
<td>0.132</td>
</tr>
<tr>
<td>Cd 1.576</td>
<td></td>
<td>1.300</td>
</tr>
<tr>
<td>St 3.0</td>
<td>0.130</td>
<td>0.125</td>
</tr>
<tr>
<td>Cd 1.551</td>
<td></td>
<td>1.275</td>
</tr>
<tr>
<td>St 4.0</td>
<td>0.120</td>
<td>0.121</td>
</tr>
<tr>
<td>Cd 1.560</td>
<td></td>
<td>1.286</td>
</tr>
</tbody>
</table>

Fig. 4: Drag and lift coefficients for a=1
Figure 5 illustrates the vorticity contours in the near wake for each aspect ratio. The dark and light contour lines corresponding to top and bottom shear layers respectively represent opposite signs of vorticity. The shear layer instabilities in the flow...
cause vortices to shed off of opposite corners as expected. The shedding period is increased as the aspect ratio increases. This agrees with the trends in the data given in Table 3. Figure 6 shows a close up look of pressure contours for aspect ratio 1, 2, 3 and 4. The separation and reattachment characteristics are visualized with no reattachment for $a=1$ as opposed to the separated flow reattaching for $a=2, 3, 4$ respectively, affecting the St.

Fig 6: Close up look of pressure contours for aspect ratio a) $a=1$; b) $a=2$; (c) $a=4$; (d) $a=4$
4.4.2 Multiple Bluff Bodies

The second series of test presented test the LB model in the application of multiple bluff body cases. The first model reproduces qualitatively compare flow pattern visualizations around building complex of Murakami (1990). The present simulations were carried out on a 600×600 computational grid. A free stream velocity corresponding to a wind originating from the south south-west direction is prescribed on the left and bottom boundaries. A constant pressure, zero-velocity gradient is applied to the top and right boundaries. Figure 7 illustrates the flow field past the array of buildings. Figure 7a shows the instantaneous streamlines of the flow indicating low velocities in the near wake of the buildings as expected. These velocity patterns are similar to those of Murakami. Figure 7b shows the corresponding vorticity contours lines, showing separation points occurring at the sharp corners with opposite signs of vorticity on opposite corners, as expected.

Fig. 7: 2-D flow past multiple bluff bodies a) streamlines and b) vorticity contour lines.
At this stage we are confident in developing new multiple bluff body studies at $Re \leq 250$ with the purpose of reducing bluff body forces. The single square bluff body presented above is amended by placing two smaller bluff bodies in front and in the near wake respectively. Figure 8 compares the flow patterns to the single bluff-body studies. For this multiple bluff body study the same computational domain and similar boundary conditions are prescribed. Flow patterns were compared to the results of first case study. Figure 8 shows the instantaneous streamlines for the (a) single square cylinder and multiple square cylinders with (b) inflow disturbance and (c) near wake disturbance. Figure 9 shows the instantaneous vorticity contours at two different time steps for each of the cases in Figure 8. Changes in the wake pattern can be observed for both inflow and near wake disturbance. A larger increase in period can be observed for inflow disturbance with a smaller increase for near wake disturbance.

Fig. 8: Streamlines for (a) undisturbed wake flow and disturbance of wake flow. (b) Inflow and (c) near wake.
Fig. 9: Vorticity contours (a) undisturbed wake flow and disturbance of wake flow. (b) Inflow and (c) near wake.
Figure 10 illustrates the vertical velocity components at the same point in the wake. It further illustrates the changes in near wake velocities due to flow disturbances caused by the smaller bluff bodies relative to no flow obstructions. One can observe changes in cross-flow forces and shedding frequencies relative to the single bluff body case (St=0.14). Inflow disturbances cause lower shedding frequency (St=0.08) in agreement with the larger period in figure 9b. The magnitude of velocities is slightly smaller which results in a smaller reduction in cross-flow forces. Wake disturbances cause smaller changes in shedding frequency (St=0.13) which agree with the smaller change in period in figure 9c. The magnitude of velocities is also smaller which results in a larger reduction in cross flow forces, as expected.

Fig. 10: Near wake velocity time histories - , fig. 9a; -.-, fig. 9b; --, fig.9c
In this thesis, a 2-D Lattice Boltzmann (LB) solver was developed and results are validated for flow past rectangular square cylinders at low Reynolds numbers (< 250). The solver was then applied to multiple bluff body cases and results presented. The LB method is an alternative computational fluid dynamics (CFD) method used to numerically predict incompressible viscous flow. The current LB method uses a statistical mechanics formulation to solve the Boltzmann equation.

The presented LB simulation been shown to capture some key features in vortex dynamics. Results were found to agree well for flow around a single square cylinder for \( \text{Re} \leq 250 \). The simulations captures the qualitative effects of increasing aspect ration such as a decrease in both \( C_d \) and St which is to be expected, however some predicted values such as lift and drag coefficients were slightly higher than other investigators. This over prediction is believed to be contributed low grid resolution in the boundary layer as a result of the uniform grid. The model was then applied to multiple bluff bodies to study the flow effects of the single bluff body. A reduction in cross-flow forces whether the smaller bluff bodies are located in front or in the near wake was observed. The changes in wake patterns and characteristics due disturbances were consistent and agreed with expected theories. The vortex shedding simulations presented provided reasonable results on coarse uniform grids compared to non-uniform FD methods.
REFERENCES


APPENDIX A: CHAMPMAN ENSKOG EXPANSION

In order to derive the Navier Stokes equations from the LB equation, the Chapman Enskog expansion is used. The Chapman Enskog expansion corresponds to a multi-scale expansion to first order in space and second order in time. The time and space coordinates are rescaled as

\[ t_1 = \varepsilon t, \quad t_2 = \varepsilon^2 t, \quad x_1 = \varepsilon x, \]

and the corresponding Taylor expansion of the derivative are

\[ \frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2}, \quad \frac{\partial}{\partial x} = \varepsilon \frac{\partial}{\partial x_1}. \]

The Chapman Enskog expansion assumes that the diffusion time scale, \( t_2 \), is much slower than the convection time scale \( t_1 \). The single particle distribution function, \( n_\alpha \), is expanded about equilibrium as

\[ n_\alpha = n^{(0)}_\alpha + \varepsilon n^{(1)}_\alpha + \varepsilon^2 n^{(2)}_\alpha + O(\varepsilon^3) \]

where it is note that \( n^{(0)}_\alpha = n^{eq}_\alpha \).

The Chapman Enskog expansion is applied to equation (14). Assuming the \( \Delta t \) is small and equal to \( \varepsilon \), i.e., \( \Delta t = \varepsilon \) equation (14) is expressed as

\[ n_\alpha(x + c_\alpha \varepsilon t + \varepsilon) - n_\alpha(x, t) = -\frac{1}{\tau}[n_\alpha(x, t) - n^{(eq)}_\alpha(x, t)] \]  

Taking a Taylor expansion of the first term on the left-hand side of (28) in time and space about point \((x,t)\) leads to

\[ \varepsilon \left( \frac{\partial}{\partial t} + c_\alpha \frac{\partial}{\partial x_1} \right) n_\alpha + \frac{1}{2} \varepsilon^2 \left( \frac{\partial}{\partial t} + c_\alpha \frac{\partial}{\partial x_1} \right)^2 n_\alpha + 3(\varepsilon^2) = -\frac{1}{\tau}(n_\alpha - n^{(0)}_\alpha). \]

Equation (29) to order \( \varepsilon \) is expressed as
\[
\left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(0)}_\alpha = -\frac{1}{\tau} n^{(1)}_\alpha
\]  

(30)

and to order \( \varepsilon^2 \) is expressed as

\[
\left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(1)}_\alpha + \frac{1}{2} \left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right)^2 n^{(0)}_\alpha = -\frac{1}{\tau} n^{(2)}_\alpha .
\]  

(31)

Substitution of equation (30) into (31) gives

\[
\left( 1 - \frac{1}{2\tau} \right) \left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(1)}_\alpha = -\frac{1}{\tau} n^{(2)}_\alpha
\]  

(32)

Adding equation (30) and equation (32) multiplied by \( \varepsilon \) then summing about \( \alpha \), i.e., \( \Sigma [\text{Eq. (30)} \times \varepsilon \times \text{Eq. (32)}] \) gives

\[
\sum \alpha \left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(0)}_\alpha + \varepsilon \sum \alpha \left( 1 - \frac{1}{2\tau} \right) \left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(1)}_\alpha = -\frac{1}{\tau} \sum \alpha \left( n^{(1)}_\alpha + \varepsilon n^{(2)}_\alpha \right)
\]

which taken to first order simplifies to

\[
\frac{\partial}{\partial t} \left( \sum \alpha n^{(0)}_\alpha \right) + c_{al} \frac{\partial}{\partial x_i} \left( \sum \alpha n^{(0)}_\alpha \right) = 0
\]  

(33)

Inserting the macroscopic quantities by applying equations given in (18), the continuity equation is recovered.

\[
\frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x_i} \rho u_i = 0
\]  

(34)

Adding equation (30) and equation (32) multiplied by \( \varepsilon \) then multiplying by \( c_{al} \) summing about \( \alpha \), i.e., \( \Sigma c_{al} \left[ \text{Eq. (30)} \times \varepsilon \times \text{Eq. (32)} \right] \) gives

\[
\sum \alpha c_{al} \left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(0)}_\alpha + \varepsilon \sum \alpha c_{al} \left( 1 - \frac{1}{2\tau} \right) \left( \frac{\partial}{\partial t} + c_{al} \frac{\partial}{\partial x_i} \right) n^{(1)}_\alpha = -\frac{1}{\tau} \sum \alpha c_{al} \left( n^{(1)}_\alpha + \varepsilon n^{(2)}_\alpha \right)
\]
which taken to first order simplifies to

\[
\frac{\partial}{\partial t} \left( \sum_\alpha c_{a\alpha} n_\alpha^{(0)} \right) + \frac{\partial}{\partial x_m} \left( \sum_\alpha c_{a\alpha} c_{am} n_\alpha^{(0)} \right) + \varepsilon \left( 1 - \frac{1}{2\tau} \right) \frac{\partial}{\partial x_m} \left( \sum_\alpha c_{a\alpha} c_{am} n_\alpha^{(1)} \right) = 0 \quad (35)
\]

Inserting the macroscopic quantities by applying equations given in (18), the momentum equation is recovered

\[
\frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x_i} \left( \Pi_{lm} + \pi_{lm} \right) = 0 \quad (36)
\]

where the total momentum flux tensor \((\Pi_{lm} + \pi_{lm})\) is separated into non-dissipative and dissipative contributions \(\Pi_{lm}, \pi_{lm}\) given by

\[
\Pi_{lm} = \sum_\alpha c_{a\alpha} c_{am} n_\alpha^{(0)} = \rho \delta_{lm} + \rho u_l u_m \quad (37a)
\]

\[
\pi_{lm} = \left( 1 - \frac{1}{2\tau} \right) \sum_\alpha c_{a\alpha} c_{am} n_\alpha^{(1)} = \nu \left( \frac{\partial u_l}{\partial x_m} + \frac{\partial u_m}{\partial x_l} \right) \quad (37b)
\]

The separation of the momentum flux tensor non-dissipative and dissipative contribution corresponds to separating single particle distribution function into equilibrium and non-equilibrium contributions. Equation (36) is the same as the Navier Stokes equations if the density fluctuation about the constant density \(\rho\) is small enough to be considered perturbations treatment.
APPENDIX B: LB SAMPLE CODE

The LB solver was written in Fortran 77. The program code consists of two parts, initialization and main time-stepping loop and is presented below. The code was run on a Pentium 4 3.0 GHz computer with 1 GB of RAM. On an average computational grid of 300,000 grid nodes at 40,000 time-steps, the simulation took an average of 6 hours to complete. The code is serial, but is very amenable to parallelization.

Initialization

The subroutine **read_parameters** is called first. This subroutine reads the initial parameters t_max, density, and omega which give the maximum number of time steps, density and relaxation time parameter respectively. The subroutine **read_obst** is called next. This subroutine reads the non-fluid nodes from a file or by user input by assigning the location the value true. This label is used to distinguish non-fluid nodes from fluid nodes. The subroutine **init_density** is called next. This subroutine initializes the particle distribution functions and is also used to supply initial conditions.

Main Loop

The main loop is run until t_max is reached. The integral fluid density is checked each t_max/10 iteration as a indicator whether the program is going to crash. The integral fluid density should be constant all time. The fluid-velocities u, v and pressure are computed from particle velocity distributions, and written to file. This gives the velocity and pressure fields at a given time. The subroutine **Inflow** is called first. The subroutine prescribes the inlet velocity boundary condition. A uniform constant velocity is prescribed at the first lattice column. This is accomplished by setting the zeroth lattice column to an equilibrium distribution at velocity Uo with zero y components. The
subroutine propagate is called next to implement the particle streaming step. All fluid densities are propagated from non-occupied nodes along the lattice connection lines to their next neighbors. The subroutine BC is called next. The no-slip boundary condition is enforced via the half-bounce back scheme. The velocity vector of all fluid densities is inverted, so all the fluid densities will be sent back to the node where they were located before the last propagation step, but with opposite velocity vector. The subroutine relaxation is called next. The collision step is implemented via density relaxation: a single time relaxation with relaxation parameter omega is applied. This step is only "local", and nothing is propagated through the lattice. The subroutine calc_F is called next and is used to estimate forces on the obstacle. The forces on the bluff body are now calculated via momentum exchange. The subroutine outlet is called next. The outlet boundary condition is now applied. A constant density and zero velocity gradients in the x direction are applied.
program lbm

: implicit none
: 
: cc parameters
: cc grid size in x- and y-dimension
:integer  y,lx,ly,sqx1,sqx2,sqy1,sqy2
:real*8 Uo
:parameter(lx=1000,ly=300,Uo=0.1)
:parameter(sqx1=200,sqx2=220,sqy1=140,sqy2=160)
:
: cc begin initialization
: cc read parameter file
:call read_parametrs(error,t_max,density,accel,omega,r_rey)
: cc read obstacle file
:call read_obst(obst,lx,ly,sqx1,sqx2,sqy1,sqy2)
: cc Calculation of viscosity
: visc = 1.d0 / 6.d0 * (2.d0 / omega - 1.d0)
: cc end initialization
:
: cc Begin Time-stepping Main loop
: do 100 time = 1, t_max
: cc Check integral fluid density
:if (time .ge. 20000 .and. mod(time,t_max/70) .eq. 0) then
:call check_density(lx,ly,node,time)
:call write_results(lx,ly,obst,node,density,time,t_max)
:end if
:
: cc Inflow BC
:call Inflow(lx,ly,obst,node,n_hlp,density,omega,Uo,u1,v1)
:
: cc Streaming Step
:call propagate(lx,ly,node,n_hlp)
: cc Apply BC No-Slip at solid walls Free-Slip Top and Bottom
:call BC(lx,ly,obst,node,n_hlp,sqx1,sqx2,sqy1,sqy2)
:
: cc Collision Step
:call relaxation(density,omega,lx,ly,node,n_hlp,obst,u1,v1)
: cc Outlet BC
:call outlet(lx,ly,obst,node,n_hlp,density,omega,Uo,u1,v1)
:
: cc Calculate Forces
:if(time.ge.10000 .and. mod(time,10).eq.0) then
:call calc_F(lx,ly,density,node,force,sqx1,sqx2,sqy1,sqy2)
:endif
: cc end of the main loop
: 100 continue
Output Flow Field:
call write_results(lx,ly,obst,node,density,time,t_max)
:end

---

subroutine read_param(t_max,density,accel,omega,r_rey)
:implicit none
:real*8  density,accel,omega,r_rey
:integer  t_max
:logical  error
:
:open(10,file='lbm.par')
:cc number of iterations
:read(10,*,err=900) t_max
:cc reference fluid density
:read(10,*,err=900) density
:cc relaxation parameter
:read(10,*,err=900) omega
:close(10)
:return
:end

subroutine read_obst(obst,lx,ly,sqx1,sqx2,sqy1,sqy2)
:implicit none
:integer  lx,ly,sqx1,sqx2,sqy1,sqy2,x,y
:logical  error,obst(0:lx+1,0:ly+1)
:cc Initialize Obstacle Array to False
:do y = 1, ly
:do x = 1, lx
:obst(x,y) = .false.
:enddo
:enddo
:
:cc Flag Solid Nodes
:do y = sqy1,sqy2
:do x = sqx1,sqx2
:obst(x,y) = .true.
:endo
:enddo
:return
:end

subroutine init_density(lx,ly,density,node)
:implicit none
:integer  lx,ly
real*8  density, node(0:8,0:lx+1,0:ly+1)
integer  x, y
real*8  t(0:2)

cc Assign weighting factors (depending on lattice geometry)
  t(0) = density * 4.d0 / 9.d0
  t(1) = density / 9.d0
  t(2) = density / 36.d0

cc loop over computational domain
  do 10 x = 0, lx+1
    do 10 y = 0, ly+1
      cc equilibrium density for zero velocity
      node(0,x,y) = t(0)
      cc equilibrium density for orthogonal velocity
      do i=1,4
        node(i,x,y) = t(1)
      enddo
      cc equilibrium densities for diagonal velocity
      do i=5,8
        node(i,x,y) = t(2)
      enddo
    10 continue
return
end

subroutine calc_F(lx,ly,density,node,force,sqx1,sqx2,sqy1,sqy2)
implicit none
integer  lx,ly,x,y,i,sqx1,sqx2,sqy1,sqy2
real*8  density,node(0:8,0:lx+1,0:ly+1),force(0:8)
real*8  t_0,t_1,t_2,Fx,Fy

cc Initialize force array to zero
  do i = 0,8
    force(i) = 0.d0
  enddo

cc Loop Over Bluff Body Edges

cc Left Edge
  x = sqx1
  do y = sqy1, sqy2
    force(3) = force(3) + 1.d0*(node(3,x,y) + node(1,x-1,y))
    force(6) = force(6) + 1.d0*(node(6,x,y) + node(8,x-1,y+1))
    force(7) = force(7) + 1.d0*(node(7,x,y) + node(5,x-1,y-1))
  enddo

cc Right Edge
x = sqx2
:do y = sqy1, sqy2
:force(1) = force(1) + 1.d0*(node(1,x,y) + node(3,x+1,y))
:force(5) = force(5) + 1.d0*(node(5,x,y) + node(7,x+1,y+1))
:force(8) = force(8) + 1.d0*(node(8,x,y) + node(6,x+1,y-1))
:endo
d:
c
:cc Top Edge
:y = sqy2
:do x = sqx1, sqx2
:force(2) = force(2) + 1.d0*(node(2,x,y) + node(4,x,y+1))
:force(5) = force(5) + 1.d0*(node(5,x,y) + node(7,x+1,y+1))
:force(6) = force(6) + 1.d0*(node(6,x,y) + node(8,x-1,y+1))
:endo
d:
c
:cc Bottom Edge
:y = sqy1
:do x = sqx1, sqx2
:force(4) = force(4) + 1.d0*(node(4,x,y) + node(2,x,y-1))
:force(7) = force(7) + 1.d0*(node(7,x,y) + node(5,x-1,y-1))
:force(8) = force(8) + 1.d0*(node(8,x,y) + node(6,x+1,y-1))
:endo
d:
c
:cc Resolve X and Y component of Force
:Fx = force(3) - force(1) + 0.7071*(force(6) + force(7) - force(8) - force(5))
:Fy = force(4) - force(2) + 0.7071*(force(7) + force(8) - force(5) - force(6))
:
:return
:end

subroutine check_density(lx,ly,node,time)

implicit none
integer lx,ly,time,x,y,n
real*8 node(0:8,0:lx+1,0:ly+1),n_sum
:n_sum = 0.d0
:cc Loop over computational domain
:do 10 y = 2, ly-1
:do 10 x = 1, lx
:cc Loop over all densities
:do 10 n = 0, 8
:10 n_sum = n_sum + node(n,x,y)
:
write(6,*) '*** Iteration number = ', time
write(6,*) '*** Integral density = ', n_sum

48
subroutine Inflow(lx,ly,obst,node,n_hlp,density,omega,Uo,u1,v1)
implicit none
integer lx,ly,i,j,x,y
logical obst(0:lx+1,0:ly+1)
real*8 node(0:lx+1,0:ly+1),density,omega,Uo,u1(0:ly+1)
real*8 c_squ,t_0,t_1,t_2,u_x,u_y,u_n(8),n_equ(0:8),u_squ,d_loc
real*8 n_hlp(0:8,0:lx+1,0:ly+1),v1(0:ly+1)
:t(0) = 4.d0 / 9.d0
t(1) = 1.d0 / 9.d0
t(2) = 1.d0 / 36.d0
c_squ = 1.d0 / 3.d0
d_loc = density
u_x = Uo
u_y = 0.d0
u_squ = u_x * u_x + u_y * u_y
u_n(1) = u_x
u_n(2) = u_y
u_n(3) = - u_x
u_n(4) = - u_y
u_n(5) = u_x + u_y
u_n(6) = - u_x + u_y
u_n(7) = - u_x - u_y
u_n(8) = u_x - u_y
cc Calc Equilibrium
n_equ(0) = t_0 * d_loc* (1.d0 - u_squ / (2.d0 * c_squ))
do i=1,4
n_equ(i) = t(1) * d_loc* (1.d0 + u_n(i) / c_squ +
u_n(i)**2.d0 / (2.d0 * c_squ**2.d0)
:- u_squ / (2.d0 * c_squ))
:enndo
do i=5,8
n_equ(i) = t(2) * d_loc* (1.d0 + u_n(i) / c_squ +
u_n(i)**2.d0 / (2.d0 * c_squ**2.d0)
:- u_squ / (2.d0 * c_squ))
:enndo
cc Set distribution function to equilibrium
do y = 0, ly+1
do i = 0, 8
subroutine outlet(lx,ly,obst,node,n_hlp,density,omega,Uo,u1,v1)
    implicit none
    integer  lx,ly,i,j,x,y
    logical  obst(0:lx+1,0:ly+1)
    real*8  node(0:8,0:lx+1,0:ly+1),density,omega,Uo,u1(0:ly+1)
    real*8  c_squ,t_0,t_1,t_2,u_x,u_y,u_n(8),n_equ(0:8),u_squ,d_loc
    real*8  n_hlp(0:8,0:lx+1,0:ly+1),v1(0:ly+1)
    : 
    : do y= 1,ly
    :   d_loc=density
    :   u_x = u1(y)
    :   u_y = v1(y)
    :   u_squ = u_x * u_x + u_y * u_y:
    :   u_n(1) =   u_x
    :   u_n(2) =         u_y
    :   u_n(3) = - u_x
    :   u_n(4) =       - u_y
    :   u_n(5) =   u_x + u_y
    :   u_n(6) = - u_x + u_y
    :   u_n(7) = - u_x - u_y
    :   u_n(8) =   u_x - u_y
    : 
    : cc Calc Equilibrium
    : n_equ(0) = t_0 * d_loc* (1.d0 - u_squ / (2.d0 * c_squ))
    : do i=1,4
    : n_equ(i) = t(1) * d_loc*(1.d0 + u_n(i) / c_squ +
    : u_n(i)** 2.d0 / (2.d0 * c_squ ** 2.d0)
    : - u_squ / (2.d0 * c_squ))
    : enddo
    : do i=5,8
    : n_equ(i) = t(2) * d_loc*(1.d0 + u_n(i) / c_squ +
    : u_n(i)** 2.d0 / (2.d0 * c_squ ** 2.d0)
:- u_squ / (2.d0 * c_squ))
:endo
:
:do i=0,8
:node(i,lx,y)=n_equ(i)
:endo
:endo
:
:return
:end

subroutine BC(lx,ly,obst,node,n_hlp,sqx1,sqx2,sqy1,sqy2)
:implicit none
:integer   lx,ly,i,sqx1,sqx2,sqy1,sqy2,x,y
:logical   obst(0:lx+1,0:ly+1)
:real*8    node(0:8,0:lx+1,0:ly+1),n_hlp(0:8,0:lx+1,0:ly+1)
 :
:do x = 1, lx
:do y = 2, ly-1
 :
:if (obst(x,y)) then
  cc Rotate all densities and write back to node
 :node(1,x,y) = n_hlp(3,x,y)
 :node(2,x,y) = n_hlp(4,x,y)
 :node(3,x,y) = n_hlp(1,x,y)
 :node(4,x,y) = n_hlp(2,x,y)
 :node(5,x,y) = n_hlp(7,x,y)
 :node(6,x,y) = n_hlp(8,x,y)
 :node(7,x,y) = n_hlp(5,x,y)
 :node(8,x,y) = n_hlp(6,x,y)
 :
:endif
:endo
 :
: return
: end

subroutine propagate(lx,ly,node,n_hlp)
:implicit none
:integer   lx,ly,i,j,x,y,x_e,x_w,y_n,y_s
:real*8    node(0:8,0:lx+1,0:ly+1),n_hlp(0:8,0:lx+1,0:ly+1)
 :
 :do j = ly,1,-1
 :do i = 1, lx
  n_hlp(2,i,j) = node(2,i,j-1)
  n_hlp(6,i,j) = node(6,i+1,j-1)

51
*:n_hlp(0,i,j) = node(0,i,j)*
*:enddo*
*:enddo*  
*:do j = ly,1,-1*  
*:do i = lx,1,-1*  
*:n_hlp(1,i,j) = node(1,i-1,j)*  
*:n_hlp(5,i,j) = node(5,i-1,j-1)*  
*:enddo*  
*:enddo*  
*:do j = 1,ly*  
*:do i = lx,1,-1*  
*:n_hlp(4,i,j) = node(4,i,j+1)*  
*:n_hlp(8,i,j) = node(8,i-1,j+1)*  
*:enddo*  
*:enddo*  
*:do j = 1,ly*  
*:do i = 1,lx*  
*:n_hlp(3,i,j) = node(3,i+1,j)*  
*:n_hlp(7,i,j) = node(7,i+1,j+1)*  
*:enddo*  
*:enddo*  
*:return*  
*:end*  

**subroutine relaxation**(density,omega,lx,ly,node,n_hlp,obst,u1,v1,h)  
*:implicit none*  
*:integer  lx,ly,x,y,i*  
*:logical  obst(0:lx+1,0:ly+1)*  
*:real*8 density,omega,node(0:8,0:lx+1,0:ly+1)*  
*:real*8 n_hlp(0:8,0:lx+1,0:ly+1),h,force(0:8)*  
*:real*8 u1(0:ly+1),v1(0:ly+1)*  
*:real*8  c_squ,t_0,t_1,t_2,u_x,u_y,u_n(8),n_equ(0:8),u_squ,d_loc*  
*:  cc Loop over computational domain*  
*:  do x = 1, lx*  
*:  do y = 1, ly*  
*:    if (.not. obst(x,y)) then*  
*:      d_loc = 0.d0*  
*:      do i = 0, 8*  
*:        d_loc = d_loc + n_hlp(i,x,y)*  
*:      enddo*  
*:    endif*  
*:  enddo*  
*:  return*  
*:end
cc x-, and y- velocity components

\[ u_x = \frac{(n\_hlp(1,x,y) + n\_hlp(5,x,y) + n\_hlp(8,x,y) - (n\_hlp(3,x,y) + n\_hlp(6,x,y) + n\_hlp(7,x,y)))}{d\_loc} \]

if(x.eq.lx-1) then
  \[ u1(y) = u_x \]
endif

\[ u_y = \frac{(n\_hlp(2,x,y) + n\_hlp(5,x,y) + n\_hlp(6,x,y) - (n\_hlp(4,x,y) + n\_hlp(7,x,y) + n\_hlp(8,x,y)))}{d\_loc} \]

if(x.eq.lx-1) then
  \[ v1(y) = u_y \]
endif

\[ u\_squ = u_x * u_x + u_y * u_y \]

cc Calc Equilibrium

\[ n\_equ(0) = t_0 \times d\_loc \times (1.d0 - u\_squ / (2.d0 * c\_squ)) \]

: do i=1,4
  \[ n\_eq(i) = t(1) \times d\_loc \times (1.0 + u\_n(i)/c\_sq + u\_n(i)**2.0/(2.0*c\_sq** 2.d0)-u\_sq/(2.0*c\_sq)) \]
:enddo

: do i=5,8
  \[ n\_eq(i) = t(2) \times d\_loc \times (1.d0+u\_n(i)/c\_sq+u\_n(i)** 2.0/(2.0*c\_sq** 2.0)-u\_sq/(2.0*c\_sq)) \]
:enddo

cc Relaxation step

: do i = 0, 8
  \[ node(i,x,y) = n\_hlp(i,x,y) + omega \times (n\_equ(i) - n\_hlp(i,x,y)) \]
:enddo

: end if
: enddo
: enddo
: return
: end
VITA

Kevin Tubbs was born and raised in Baton Rouge, Louisiana. He is the son of John and Veronica Tubbs. Kevin attended Southern University Laboratory School where he participated in many athletic and academic activities. He graduated Valedictorian and chose to attend Southern University, Baton Rouge. While at Southern, Kevin participated in the national awarded Timbuktu Academy and won various awards and scholarships, including the Lawrence Livermore National Laboratory/ National Society of Black Physicist student of the year award and scholarship. Kevin received his Bachelor of Science degree in May 2001, where he graduated Magna Cum Laude. Kevin then entered the graduate program in the Department of Physics at Louisiana State University, Baton Rouge where he received a Board of Regents Fellowship and will receive a Master of Science degree in physics on May 21, 2004.

Within the study of physics and engineering, he is interested in computational modeling, computational fluid dynamics, scientific visualization and high performance computing. In the future, Kevin plans to pursue a doctoral degree in engineering sciences from LSU studying nonlinear dynamics in application to free surface breaking waves.