Contract Number: DE-FG02-03ER83715

DOE SBIR Phase II

Final Report

Development of a Prototype Lattice Boltzmann Code for CFD of Fusion Systems

Martin J. Pattison

Kannan N. Premnath

Sanjoy Banerjee

Vinay Dwivedi

MetaHeuristics LLC
209 W. Alamar Ave, Suite A
Santa Barbara
CA 93105

Abstract

Designs of proposed fusion reactors, such as the ITER project, typically involve the use of liquid metals as coolants in components such as heat exchangers, which are generally subjected to strong magnetic fields. These fields induce electric currents in the fluids, resulting in magnetohydrodynamic (MHD) forces which have important effects on the flow. The objective of this SBIR project was to develop computational techniques based on recently developed lattice Boltzmann techniques for the simulation of these MHD flows and implement them in a computational fluid dynamics (CFD) code for the study of fluid flow systems encountered in fusion engineering.

The code developed during this project, solves the lattice Boltzmann equation, which is a kinetic equation whose behaviour represents fluid motion. This is in contrast to most CFD codes which are based on finite difference/finite volume based solvers. The lattice Boltzmann method (LBM) is a relatively new approach which has a number of advantages compared with more conventional methods such as the SIMPLE or projection method algorithms that involve direct solution of the Navier–Stokes equations. These are that the LBM is very well suited to parallel processing, with almost linear scaling even for very large numbers of processors. Unlike other methods, the LBM does not require solution of a Poisson pressure equation leading to a relatively fast execution time. A particularly attractive property of the LBM is that it can handle flows in complex geometries very easily. It can use simple rectangular grids throughout the computational domain – generation of a body-fitted grid is not required.

A recent advance in the LBM is the introduction of the multiple relaxation time (MRT) model; the implementation of this model greatly enhanced the numerical stability when used in lieu of the single relaxation time model, with only a small increase in computer time. Parallel processing was implemented using MPI and demonstrated the ability of the LBM to scale almost linearly.

The equation for magnetic induction was also solved using a lattice Boltzmann method. This approach has the advantage that it fits in well to the framework used for the hydrodynamic equations, but more importantly that it preserves the ability of the code to run efficiently on parallel architectures. Since the LBM is a relatively recent model, a number of new developments were needed to solve the magnetic induction equation for practical problems. Existing methods were only suitable for cases where the fluid viscosity and the magnetic resistivity are of the same order, and a preconditioning method was used to allow the simulation of liquid metals, where these properties differ by several orders of magnitude. An extension of this method to the hydrodynamic equations allowed faster convergence to steady state. A new method of imposing boundary conditions using an extrapolation technique was derived, enabling the magnetic field at a boundary to be specified. Also, a technique by which the grid can be stretched was formulated to resolve thin layers at high imposed magnetic fields, allowing flows with Hartmann numbers of several thousand to be quickly and efficiently simulated.

In addition, a module has been developed to calculate the temperature field and heat transfer. This uses a total variation diminishing scheme to solve the equations and is again very amenable to parallelisation. Although, the module was developed with thermal modelling in mind, it can also be applied to passive scalar transport. The code is fully three dimensional and has been applied to a wide variety of cases, including both laminar and turbulent flows. Validations against a series of canonical problems involving both MHD effects and turbulence have clearly demonstrated the ability of the LBM to properly model these types of flow.

As well as applications to fusion engineering, the resulting code is flexible enough to be applied to a wide range of other flows, in particular those requiring parallel computations with many processors. For example, at present it is being used for studies in aerodynamics and acoustics involving flows at high Reynolds numbers. It is anticipated that it will be used for multiphase flow applications in the near future.

Contents

Summary

The objective of this Phase II SBIR project was to develop a computational fluid dynamics (CFD) code to simulate three dimensional fluid dynamics problems in complex geometries, with an emphasis on modelling the types of flow of relevance to fusion engineering. In particular, liquid metals are used as coolants in many designs, and magnetohydrodynamic (MHD) effects due to high magnetic field strengths need to be properly dealt with for application to these designs.

The code developed during this project, MetaFlow, uses the lattice Boltzmann method (LBM) to solve the equations governing the fluid motion. This is in contrast to most CFD codes which are based on finite difference/finite volume based solvers. The LBM is a relatively new approach which has been widely developed and investigated in research institutions and universities, and has a number of advantages compared with more conventional methods such as the SIMPLE or projection method algorithms. These are that the LBM is much easier to parallelize and is much more scalable for very large problems. This is because LB methods only require that a grid point share data with a small number of surrounding neighbours during a computation, whereas equivalent continuum CFD methods, (e.g. the widely used projection method of Temam, 2001) involve solution of an elliptic Poisson-type equation that requires extensive non-local data. Furthermore, solution of the Poisson equation takes up 80-90% of CPU time. As this is not required in the LBM, MetaFlow is much faster per time step. A particularly attractive property of the LBM is that it can handle flows in complex geometries very easily. It can use a simple rectangular grid throughout the computational domain – generation of a body-fitted grid is not required. This is because the LBM maintains high accuracy at cut-cell boundaries in a Cartesian grid, eliminating the need for the complex boundary-fitted grids required in a conventional CFD code.

A number of variants of the LBM are available. Most codes, including the only commercial LBM code, Exa Corporation's PowerFlow, use the single relaxation time (SRT) model. However, a recent development is the multiple relaxation time (MRT) scheme. This is a much more advanced model which exhibits far better stability and better accuracy than the SRT. For this reason an MRT-based scheme was formulated and implemented in MetaFlow, and successfully tested for a variety of cases, including large eddy simulation (LES) of turbulence, using a Smagorinsky model with wall-damping for the unresolved scales of motion. Investigation of the performance of the code revealed the following:

- The MRT was more stable than the SRT by a factor of about three (on the basis of the maximum flow rate for a particular problem)
- Comparisons of execution speed showed MetaFlow to be faster than a finite difference based code by a factor of about three (on a per time step per grid point basis)
- Almost linear scaling could be achieved on a parallel supercomputer with distributed memory

These findings were in keeping with those found by other researchers in the field. The superior stability of the MRT makes it much easier to simulate more complex flows, particularly turbulent cases. It also avoids certain unphysical behaviour found with the SRT which could be the reason why LBM has not received wider acceptance to date, in particular for turbulence simulations.

The LBM also allows complex flows and physics to be incorporated and models for MHD have been developed and included in the MetaFlow code. The MHD solver uses a similar lattice-Boltzmann type approach to the flow solver, and this fits in naturally to the code. Also, by using an LB approach for the MHD preserves the ability of MetaFlow to run efficiently on parallel architectures. Very little work had previously been done on LBMs for MHD, and whilst existing models from the literature were adequate for some simple flows and problems of academic interest, they proved unsatisfactory for application to more general problems of interest to fusion engineering, in particular. A number of new innovations were needed, and these included:

- Existing models were restricted to flows with magnetic Prandtl numbers, Pr_m, of order one. In fusion applications, Pr_m is typically several orders of magnitude lower. In response to this, a method based on a preconditioning algorithm with an adjustable scaling parameter has been developed to enable the simulation of flows with low Pr_m. This model also allows more rapid convergence to steady-state flows.

- A new method of imposing boundary conditions has been formulated. This involves using an extrapolation technique and enables the magnetic field at a boundary to be specified.

- A technique by which the grid can be stretched was formulated to resolve thin layers at high imposed magnetic fields. *This allowed flows with Hartmann numbers of several thousand to be quickly and efficiently simulated.*

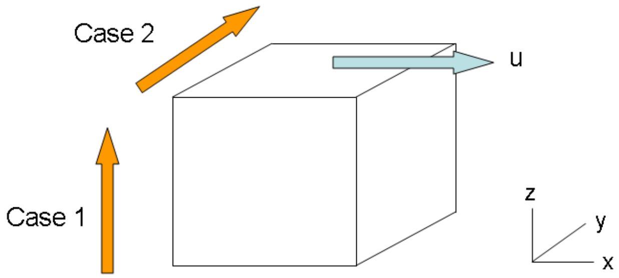
The approach was found to be efficient in dealing with three-dimensional high Hartmann (Ha) number MHD flow problems, an area where conventional approaches have significant limitations.

In addition, a module has been developed to calculate the temperature field and heat transfer. This uses a total variation diminishing (TVD) scheme to solve the equations and is again very amenable to parallelisation. Although, the module was developed with thermal modelling in mind, it can also be applied to passive scalar transport.

In support of the code, a graphical user interface (GUI) has been written so that the user can easily input the problem specification. Also, a module for reading and processing CAD files was incorporated, in order that complex geometries could easily be imported into the code. Also, the GUI links to MatLab, allowing rapid visualisation of the results.

An extensive set of validation and other simulations have been performed, and some of the key results are summarized . For turbulence modelling, these have included LES of channel flow with a free surface and the flow through a square duct. Comparison with benchmark results showed very good agreement, even for the more challenging case of the duct flow, where the characteristic secondary flows were correctly predicted.

For MHD flows, the 3-D lid driven cavity provides an excellent test of a code, since, unlike many other test cases, all the terms in the equations are non-zero, i.e. it is a fully 3D MHD problem.



Cavity is unit cube

Figure i: Lid driven cavity; orange arrows denote applied magnetic field.

Figure i shows the arrangement of this case. The flow in a cube is driven by a moving lid and a magnetic field is applied in a direction perpendicular to the motion. Figure ii shows velocity profiles calculated for Case 1 with low magnetic Reynolds number and compares them with those obtained by the MHD group at UCLA based on a finite difference method. The Hartmann number was 45 and the Reynolds number 100 (both based on length of a side of the cube). Note that the scales have been

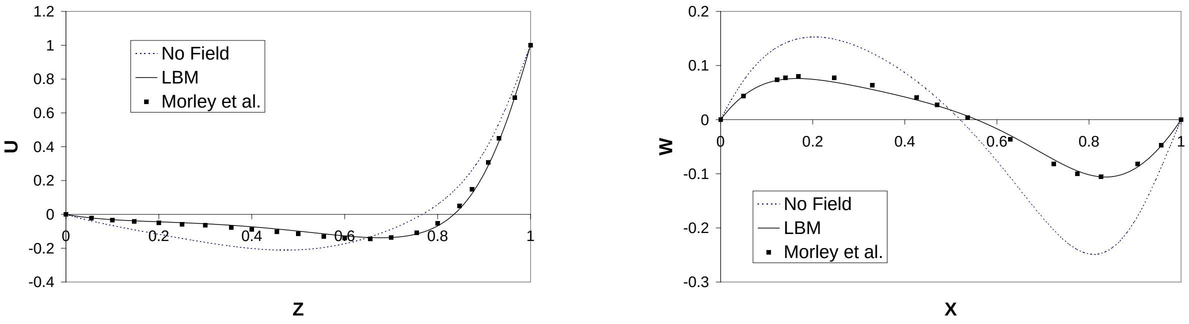


Figure ii: Velocity profiles for lid-driven cavity. Profile taken through centre of y-z plane (upper graph), centre of x-y plane (middle) and centre of x-z plane (bottom). Re=100, Ha=45. (Case 2.)

normalised such that the sides of the cube and the lid velocity are unity. Good agreement with available data is shown.

Another case done was of fully-developed turbulent flow through a pipe entering a magnetic field. The parameters were chosen to correspond to those being used in a recent experiment performed at UCLA. This computation used a large domain size of 2400×120×120 grid points and this was run in parallel on 32 processors on the DOE NERSC supercomputer Bassi. With this domain size, a length of twenty diameters was simulated, and Fig. iii shows the mean velocity profile and turbulence statistics 15 diameters into the region with the magnetic field. A Hartmann number of 20 (based on the pipe radius) and Reynolds number of 5000 (based on mean velocity and diameter) was used and the profiles are in the direction parallel to the field and through the pipe centre.

It can clearly be seen that the turbulence is greatly suppressed by the magnetic field. This is what is expected and the results were in line with preliminary results from the experimental programme. At the position where the statistics were gathered (15 diameters downstream, as in the experiment), the flow is not likely to be fully developed, and further suppression of the turbulence downstream is expected, with flattening of the velocity profile.

A series of tests were done for Hartmann flow using our new stretched grid model. Figure iv shows the velocity and induced magnetic field for a Hartmann number of 10 000 (based on half the channel width). Very good agreement between MetaFlow and the analytical solution can be seen for this case, which used 192 points. The error in the velocity at the centre was about 0.5%.

The results of the simulation of an MHD flow through a square duct are shown in Fig. v. This case had insulating walls and was run at Ha=500 – with a 128×128 grid this only took a few hours on a single processor PC. Naturally much higher Hartmann numbers could be run on parallel machines.

Two dimensional plots of field and the velocity are shown in Fig. (v). Over the central region, the velocity profile can be seen to be very flat, with the induced field having a constant gradient. These observations are in keeping with expected trends and quantitative comparisons with analytical solutions were in good agreement (as discussed in the main text).

One example involved the 3-D CFD simulation of a section of a thermal blanket. A cross section is shown schematically in Fig.vi. In this cell, there is a central region of lead-lithium (PbLi) surrounded by an insulating layer, another channel with PbLi and finally ferritic steel. The PbLi enters at a temperature of 400°C and is subjected to heating by the neutron flux, as are the other materials; the cell is 1.4m long. Figure vii shows the predicted induced magnetic field and flow velocity, and Fig. viii and ix show the

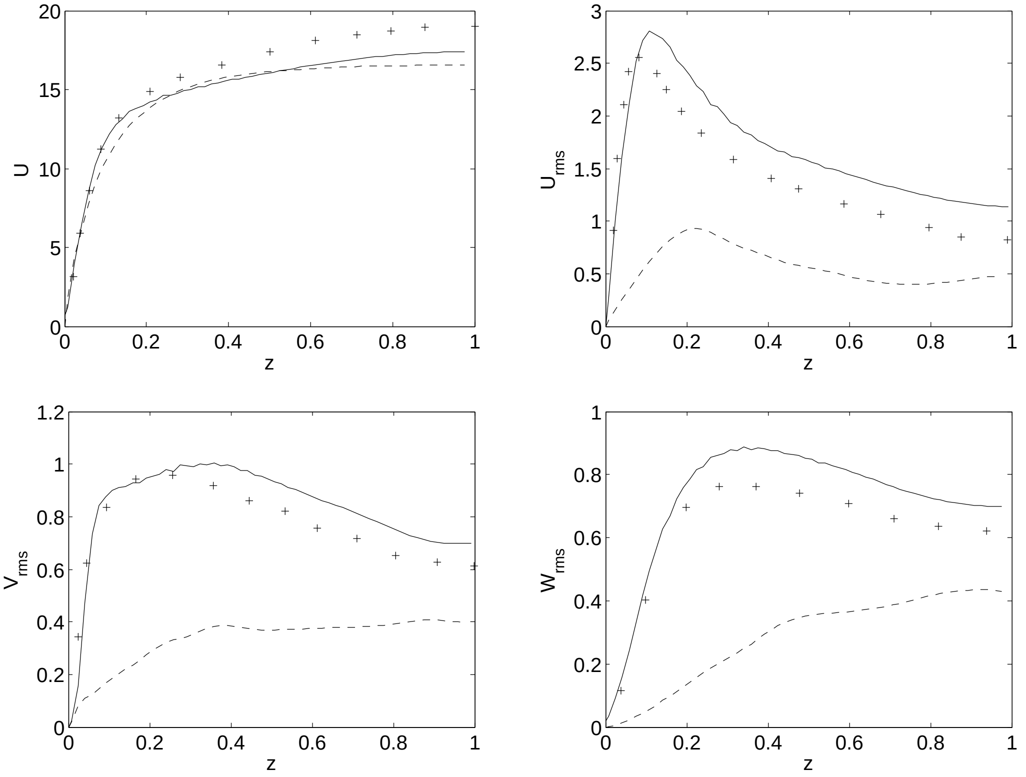


Figure iii: Profiles of mean velocity and rms fluctuations along z axis through centre. Solid line is LBM for no magnetic field, crosses DNS of Eggels et al. (1994) Ha=0 and dashed line LBM with Ha=20, Re=5000. U, V and W are streamwise, spanwise and wall-normal components, respectively, with applied field was z direction.

predicted temperature. The results are discussed more thoroughly in the main text.

The work is expected to lead to several publications. Journal articles will include one paper on the MRT model, one on a study of turbulent duct flow performed with our code, and two papers on the the advances we have made in MHD modelling. In addition, a paper on the MHD aspects was recently presented at the American Nuclear society meeting, and a paper on turbulence modelling has been accepted for a major conference on turbulent flows in Germany later this year.

In conclusion, the code has shown itself capable of accurately simulating a variety of problems, including turbulent and MHD flows and has demonstrated its efficiency on parallel machines. The code is now in a state from which it can used for studies of MHD systems of interest to fusion engineers, particularly at high Hartmann numbers in 3D, and it is hoped that it will be usefully employed in this application. In addition, the code is now being further developed with funding from NASA for aero-acoustic applications (noise from aircraft components). In the near future, it is expected that the code will be applied to some cases for the Nuclear Regulatory Commission. At present, we are forming a consortium of companies in chemical process industry to extend our code to deal with the simulation of flow and transport processes of interest to them. The lattice-Boltzmann based code – MetaFlow – in this work appears to be promising for simulating large scale 3D CFD problems of interest to the engineering community.

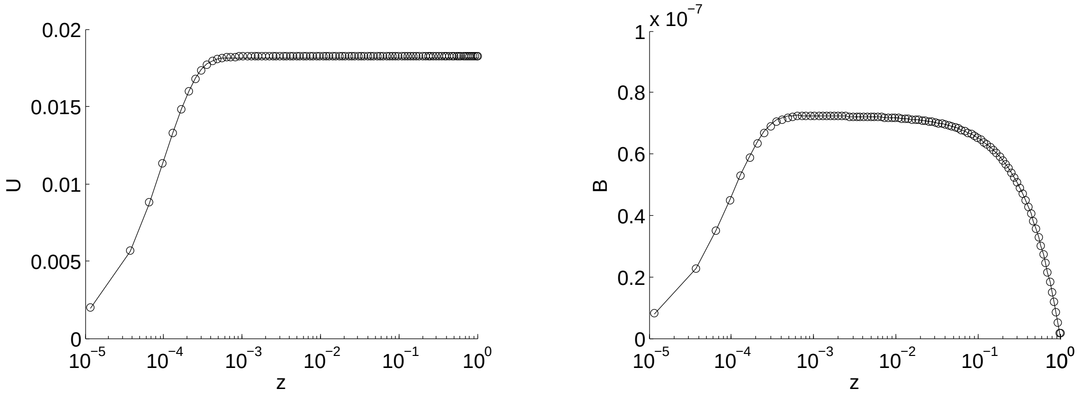


Figure iv: Velocity (U) and magnetic field (B) for Ha=10 000; line LBM, circles analytical. Abscissae are in log scales. 192 grid points.

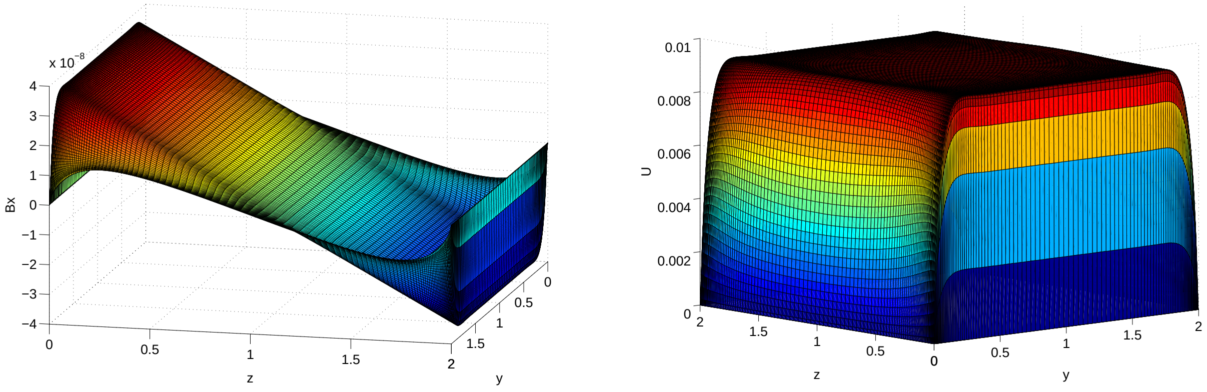


Figure v: Induced field (Bx) and velocity (U) plots for square duct case at Ha=500.

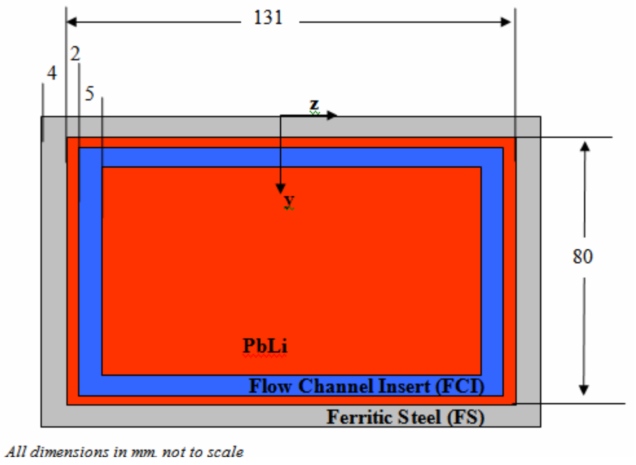


Figure vi: Schematic diagram showing cross-section of thermal blanket module.

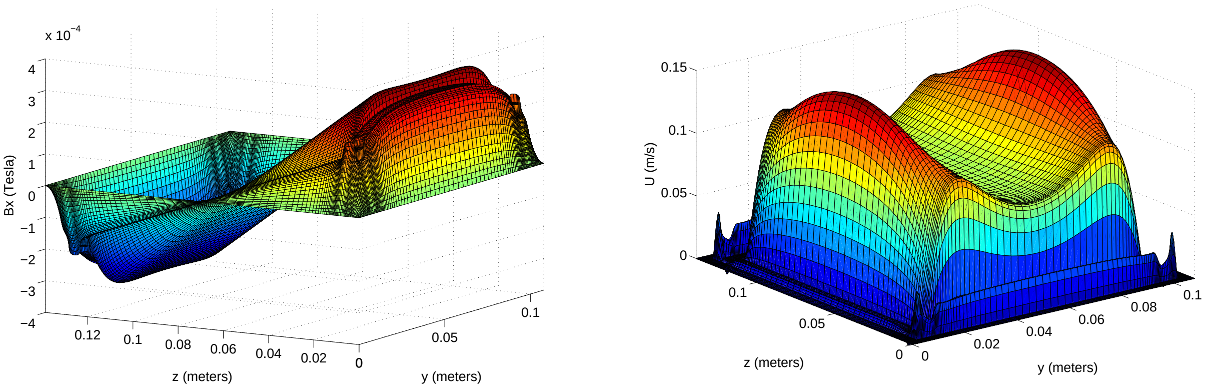


Figure vii: Induced field (Bx) and velocity (U) plots on cross-section of cell.

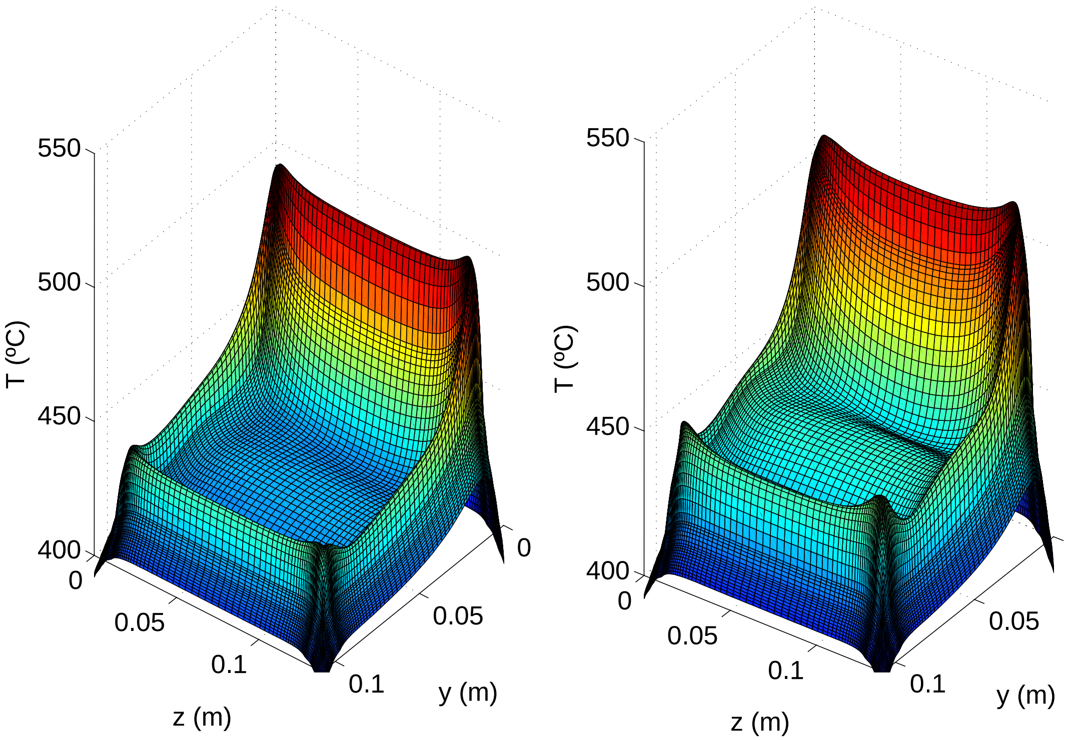


Figure viii: Temperature at 0.7m (left) and (1.4m right) from inlet.

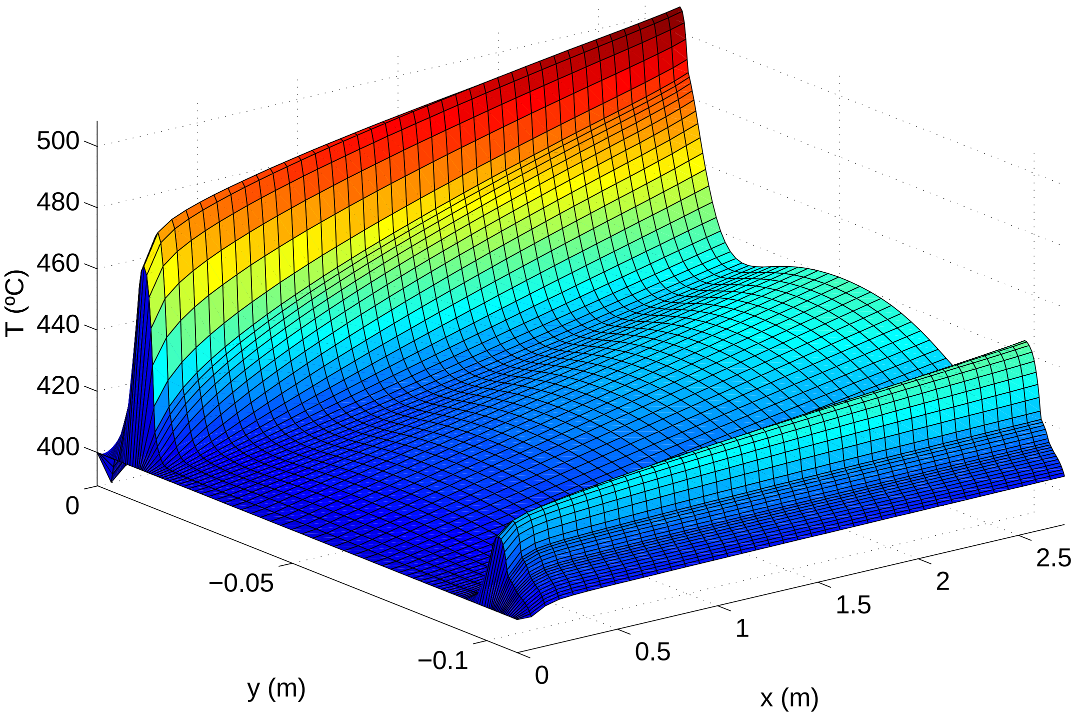


Figure ix: Temperature on $x - y$ plane through centre of duct.

Papers Arising from this Project:

Pattison, Martin J., Kannan N. Premnath and Sanjoy Banerjee (2007) Turbulence-Induced Secondary Flows in a Square Duct using a Multiple-Relaxation-Time Lattice-Boltzmann Approach. Turbulence and Shear Flow Phenomena (TSFP)-5, 27-29 August 2007, Munich, Germany.

Pattison, Martin J., Kannan N. Premnath and Neil B. Morley (2006) Lattice Boltzmann Methods for Magnetohydrodynamic Flows in Fusion Applications. American Nuclear Society (ANS) Winter Meeting, 12-16 November 2006, Albuquerque, NM

Premnath, Kannan N. and Martin J. Pattison (2007) Convergence Acceleration of Generalized Lattice Boltzmann Equation with Forcing Term through Preconditioning for Steady State Flows and Some Applications. Submitted to Phys. Rev. E.

Premnath, Kannan N., Martin J. Pattison and Sanjoy Banerjee (2007) Generalized Lattice Boltzmann Equation with Forcing Term for LES of Bounded Turbulent Flows: Accuracy, Stability and Computational Efficiency. Submitted to Physical Review E.

Pattison, Martin J., Kannan N. Premnath, Neil B. Morley and Mohamed A. Abdou (2007) Progress in Lattice Boltzmann Methods for Magnetohydrodynamic Flows relevant to Fusion Applications. Submitted to Fusion Engineering & Design.

Pattison, Martin J., Kannan N. Premnath and Sanjoy Banerjee (2006) Large Eddy Simulation of Turbulent Flow in a Square Duct using the Generalized Lattice Boltzmann Equation. Submitted to Phys. Rev. E

Pattison, Martin J. and Sanjoy Banerjee (2004) Numerical simulation of fluids using the lattice Boltzmann scheme. 6th Int. Conf. Nuclear Thermal Hydraulics, Operations and Safety (NUTHOS-6). Nara, Japan, 4-8 Oct. 2004

1 Introduction

The objective of this Phase II SBIR project was to develop a computational fluid dynamics (CFD) code to serve the needs of the fusion community. More specifically, the ITER project will use liquid metals in its heat exchangers, and computational tools to model the flow in magnetic fields where magnetohydrodynamic effects dominate will prove valuable. The code developed, MetaFlow, uses the lattice Boltzmann method (LBM) to solve the equations governing the behaviour of fluids. This is in contrast to most CFD codes which are based on finite difference/finite volume based solvers. The LBM has a number of advantages compared with more conventional methods such as the SIMPLE or projection method algorithms. These are that the LBM is much easier to parallelize and is much more scalable for very large problems. This is because LB methods only require that a grid point share data with a small number of surrounding neighbours during a computation, whereas equivalent continuum CFD methods, (e.g. the widely used projection method of Chorin, 1968 and Temam, 2001)) involve solution of an elliptic pressure Poisson equation, and require extensive non-local data. Furthermore, solution of the Poisson equation takes up 80-90% of CPU time in both our experience and that of others (e.g. Madhabushi & Vanka, 1991; Premnath et al., 2002). As this is not required in the LBM, MetaFlow is much faster per time step. A particularly attractive property of the LBM is that it can handle flows in complex geometries very easily and can use a simple rectangular grid throughout the computational domain – generation of a body-fitted grid is not required. This is because the LBM maintains high accuracy at cut-cell boundaries in a Cartesian grid, eliminating the need for the complex boundary-fitted grids needed in a conventional CFD code.

The code used the multiple relaxation time (MRT) model, a state-of-the-art lattice Boltzmann formulation which was originally developed by d'Humières et al. (2002) and extended by Premnath and Abraham (2006) and Premnath and Pattison (2007). This method has been found to have clear advantages over earlier models using a single relaxation time (SRT). The use of multiple time scales in this model results in better accuracy, and, more importantly, it is substantially more stable than SRT models for more challenging problems in CFD, particularly those involving turbulent flow. We believe that our code is set to become the first commercial code to use the MRT implementation of the LBM.

The report is organised as follows. Section 2 presents the overall elements of the code developed in this work. A particularly important aspect of the work was the simulation of liquid metal flows with magnetohydrodynamic (MHD)effects. Earlier researchers (Dellar, 2002; Breyiannis and Valougeorgis, 2004) had developed and used basic models for solving the equations of MHD within the lattice Boltzmann framework, and these were extended significantly in the work here to allow their application to practical problems. Section 3 and 4 detail the models used and the techniques developed within this project to allow their application to bounded flows and low magnetic Prandtl number fluids, representing liquid metals of interest to fusion engineering problems, and for steady state cases. A key feature of the approach taken here was again that grid points only need exchange data with nearest neighbours, allowing fast and efficient execution on parallel machines.

The potential application of MetaFlow to heat exchangers in ITER and other applications necessitated development of a suitable scheme for solving the equations governing the transport of heat. Due to the requirement of efficient execution on parallel architectures, an explicit scheme was required, with greater fidelity through control over numerical errors. Section 5 discusses the total variation diminishing (TVD) scheme which was developed and implemented in the code, that addresses this aspect.

Section 6 deals with the implementation of a stretched grid scheme within our 3D lattice Boltzmann MRT model. This proved to be computationally efficient and highly effective in resolving the thin layers characteristic of high Hartmann number flows.

MetaFlow is intended to be used for a variety of problem types, and has been designed so as to be flexible in the types of scenario it can handle and the geometries that can be modelled. Section 7

Module	Algorithm	Section
Flow solver	19 velocity Lattice Boltzmann MRT model	3.1
	Boundary conditions	3.2
	Preconditioning algorithm for steady-state	3.4
	Subgrid-scale turbulence model for LES	3.3
	MPI parallel processing	3.5
MHD model	7 velocity LBM model for magnetic induction	4
	Boundary conditions	4.2
	Low Prandtl number algorithm	4.3
	Stretched grid model for high Ha flows	6
Thermal module	TVD scheme	5

Table 1: Main functions embedded in MetaFlow code.

discusses the import of complex geometries from CAD files – more information on running the code can be found in the users manual (see www.metah.com). Section 8 presents an example of a computation of flow through a section of a thermal blanket, and the report is wrapped up with the conclusions in Section 9.

Supporting information is given in the appendices. Appendix A lists the papers arising from this project. Appendices B through E provide the details of the major components in the MRT model, with Appendix F discussing the optimised solution procedure. Appendix G discusses a method for obtaining strain rates directly from the LBM (i.e. without taking finite differences). The methodology used for accelerating convergence to steady-state is presented in Appendix H. A particularly valuable procedure developed was the extension of this method to stretched grids, and the interpolation procedure is given in Appendix I. The perturbation field used to initialise velocity fields in order that turbulence would develop is listed in Appendix J, and Appendix K presents a formulation of the magnetic induction equation in the general case of spatially varying electrical conductivity.

2 Overview of MetaFlow Code

The main functionalities of the code are illustrated schematically in Fig. 1.

The user specifies the problem to set up using a windows based graphical user interface (GUI). This GUI generates an ascii file which is then read in by MetaFlow. Since this is an ascii (i.e. human readable) file and is provided with embedded comments, it can be edited directly by the user, useful when working on remote supercomputers for which the GUI is not supported. Geometry input can be specified in two ways. One is for the user to set up a grid directly which can be imported into the code; the other is to use a CAD package to generate an STL file. The latter approach is of course the preferred method for complex geometries. The MetaFlow user manual discusses user input in more depth, and section 7 describes the import of geometry from CAD files.

The core code includes solvers for the fluid equations, magnetic induction equation that are based on LBM, and heat transfer, based on a hybrid finite difference LBM approach. Table 1 lists the principal features, along with the sections where more information can be found.

Several output files are generated by MetaFlow. These include monitor files, where parameters at requested locations are output each time step, a convergence history file, and a debug file which provides information useful to identify any problems. Files with fields such as pressure, velocity and magnetic field are generated at user-defined intervals, as are restart files. The GUI is coupled to MatLab, which enables data visualisation to be performed relatively easily using MetaFlow's standard output files.

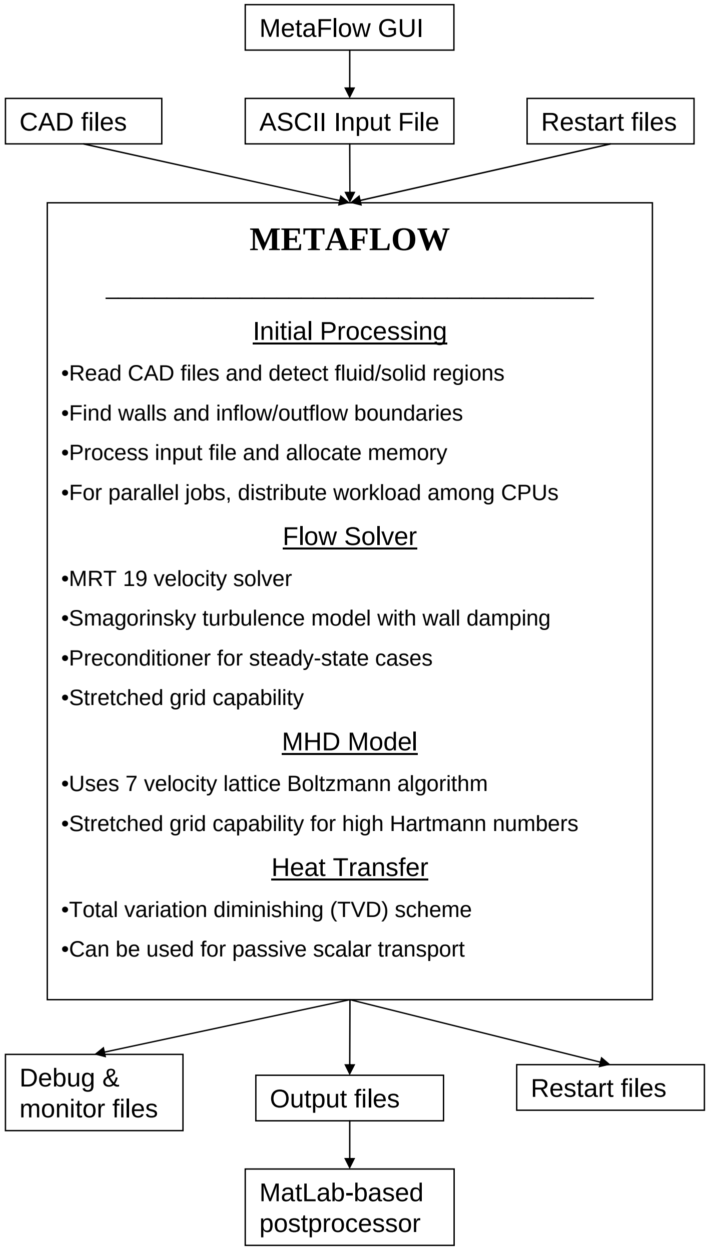

MetaFlow GUI
CAD files
ASCII Input File
Restart files
METAFLOW
Initial Processing
•Read CAD files and detect fluid/solid regions
•Find walls and inflow/outflow boundaries
•Process input file and allocate memory
•For parallel jobs, distribute workload among CPUs
Flow Solver
•MRT 19 velocity solver
•Smagorinsky turbulence model with wall damping
•Preconditioner for steady-state cases
•Stretched grid capability
MHD Model
•Uses 7 velocity lattice Boltzmann algorithm
•Stretched grid capability for high Hartmann numbers
Heat Transfer
•Total variation diminishing (TVD) scheme
•Can be used for passive scalar transport
Debug & monitor files
Output files
Restart files
MatLab-based postprocessor

Figure 1: Principal functionalities of MetaFlow lattice Boltzmann code.

As mentioned earlier, referring to Fig. 1, we now discuss the main modules including the “flow solver”, based on the lattice Boltzmann method (Section 3), the “MHD model” based on the LBM (Section 4) and the ”heat transfer model” based on the TVD scheme (Section 5). In each of these sections, validation of the methods used is included.

3 Lattice Boltzmann Model for Hydrodynamic Fields

3.1 Basic equations

The approach taken in this work is to solve the equations governing the flow of fluid using the lattice Boltzmann method (LBM). The LBM is a relatively recent approach based on kinetic theory for solving fluid mechanics and other physical problems (Succi, 2001; Chen and Doolen, 1998). In brief, the LBM consists solving the evolution equation of the distribution function f_α of particle populations as they move and collide on a lattice. In the most advanced formulation currently available, the MRT, the distribution function is calculated using the equation (d'Humières et al., 2002; Premnath and Abraham, 2006):

$$f_\alpha(\overrightarrow{x}+\overrightarrow{e_\alpha}\delta_t, t+\delta_t) - f_\alpha(\overrightarrow{x}, t) = -\sum_\beta \Lambda_{\alpha\beta}\left(f_\beta - f_\beta^{eq}\right) + \sum_\beta \left(I_{\alpha\beta} - \frac{1}{2}\Lambda_{\alpha\beta}\right) S_\beta \delta_t. \tag{1}$$

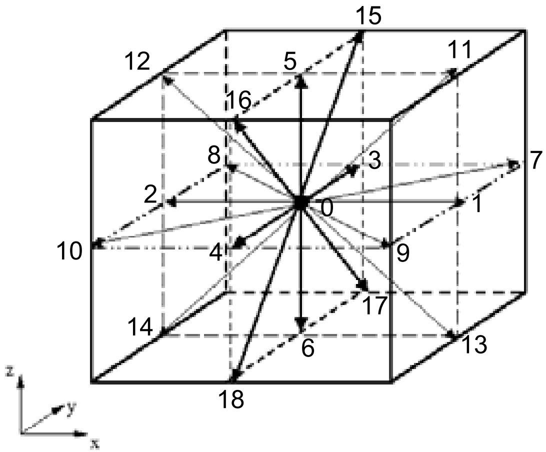


Figure 2: Three dimensional, 19 velocity (D3Q19) lattice used for LBM.

Here, the left hand side of Eq. (1) corresponds to the change in the distribution function during a time interval δ_t, as the particles stream from location $\overrightarrow{x}$ to their adjacent location $\overrightarrow{x}+\overrightarrow{e_\alpha}\delta_t$, with a velocity $\overrightarrow{e_\alpha}$ along the characteristic direction α. At this point is is worth noting that the commonly used SRT model uses a scalar relaxation parameter in place of the tensors $\Lambda_{\alpha\beta}$ and there is no summation for the terms on the right. Also note, however, that the implementation of Eq. (1) does not involve direct summation as shown, but through a highly optimised procedure that exploits certain properties of $\Lambda_{\alpha\beta}$ (see Appendix F).

We consider a three-dimensional, nineteen velocity (D3Q19) particle velocities set, shown in Fig. 2, given by

$$\overrightarrow{e_\alpha} = \begin{cases} (0,0,0) & \alpha = 1 \\ (\pm1,0,0), (0,\pm1,0), (0,0,\pm1) & \alpha = 2,\cdots,7 \\ (\pm1,\pm1,0), (\pm1,0,\pm1), (0,\pm1,\pm1) & \alpha = 8,\cdots,19, \end{cases} \tag{2}$$

The Cartesian component c of the particle velocity $\overrightarrow{e_\alpha}$ is given by $c = \delta_x/\delta_t$, where δ_t is the lattice time step. The corresponding vector of distribution functions $\mathbf{f}$ at a location may be written as

$$\mathbf{f} = \left[f_0, f_1, f_2, \ldots, f_{18}\right]^t. \tag{3}$$

The first term on the right hand side (RHS) of Eq. (1) represents the cumulative effect of particle collisions on the evolution of distribution function f_α. Collision is a relaxation process in which f_β relaxes to its local equilibrium value f_β^{eq} at a rate determined by the relaxation time matrix $\Lambda_{\alpha\beta}$. The MRT has a generalized collision matrix with multiple relaxation times corresponding to the underlying physics: The macroscopic fields such as densities, momentum and stress tensors are given as various kinetic moments of the distribution function. For example, collision does not alter the densities ρ and momentum $\overrightarrow{j} = \rho\overrightarrow{u}$, while the stress tensors relax during collisions at rates determined by fluid properties such as viscosities. The components of the collision matrix $\Lambda_{\alpha\beta}$ in the MRT are developed to reflect the underlying physics.

The second term on the RHS of Eq. (1) introduces changes in the evolution of distribution function due to external force fields $\overrightarrow{F}$ such as gravity or some driving force through a source term S_α. In this term $I_{\alpha\beta}$ is the component of the identity matrix I. The source term S_α may be written as (Premnath and Abraham, 2006)

$$S_\alpha = \frac{(e_{\alpha j} - u_j)F_j}{\rho c_s^2} f_\alpha^{eq,M}(\rho, \overrightarrow{u}), \tag{4}$$

where $f_\alpha^{eq,M}(\rho, \overrightarrow{u})$ is the local Maxwellian

$$f_\alpha^{eq,M}(\rho, \overrightarrow{u}) = \omega_\alpha \rho \left\{ 1 + \frac{\overrightarrow{e_\alpha} \cdot \overrightarrow{u}}{c_s^2} + \frac{(\overrightarrow{e_\alpha} \cdot \overrightarrow{u})^2}{2c_s^4} - \frac{1}{2}\frac{\overrightarrow{u} \cdot \overrightarrow{u}}{c_s^2} \right\}, \omega_\alpha = \left\{ \begin{array}{ll} \frac{1}{3} & \alpha = 1 \\ \frac{1}{18} & \alpha = 2, \cdots, 7 \\ \frac{1}{36} & \alpha = 8, \cdots, 19. \end{array} \right. \tag{5}$$

and $c_s = 1/\sqrt{3}c$ is the speed of sound of the model. By neglecting terms of the order of $O(Ma^2)$ or higher in its in Eq.(4) it may be simplified as (Premnath and Abraham, 2006)

$$\begin{aligned} S_\alpha \quad = \quad & w_\alpha \left[\frac{3}{c^2}(e_{\alpha x} - u_x) + \frac{9}{c^4}(\overrightarrow{e_\alpha} \cdot \overrightarrow{u})\, e_{\alpha x} \right] F_x + \\ & w_\alpha \left[\frac{3}{c^2}(e_{\alpha y} - u_y) + \frac{9}{c^4}(\overrightarrow{e_\alpha} \cdot \overrightarrow{u})\, e_{\alpha y} \right] F_y + \\ & w_\alpha \left[\frac{3}{c^2}(e_{\alpha z} - u_z) + \frac{9}{c^4}(\overrightarrow{e_\alpha} \cdot \overrightarrow{u})\, e_{\alpha z} \right] F_z \end{aligned} \tag{6}$$

where F_x, F_y and F_z are the Cartesian components of the force field.

Note that for convenience, Eq. (1) may be considered as being composed of two steps, *viz. collision* and *streaming steps*, which may be given as:

$$\tilde{f}_\alpha(\overrightarrow{x}, t) = -\sum_\beta \Lambda_{\alpha\beta}\left(f_\beta - f_\beta^{eq}\right) + \sum_\beta \left(I_{\alpha\beta} - \frac{1}{2}\Lambda_{\alpha\beta} \right) S_\beta \delta_t. \tag{7–a}$$

$$f_\alpha(\overrightarrow{x} + \overrightarrow{e_\alpha}\delta_t, t + \delta_t) - f_\alpha(\overrightarrow{x}, t) \tag{7–b}$$

respectively. In Eq. (7–a), $\tilde{f}_\alpha$ refers to the post-collision distribution function.

The local macroscopic density and velocity fields are given by

$$\rho = \sum_\alpha f_\alpha, \tag{8}$$

$$\overrightarrow{j} = \rho\overrightarrow{u} = \sum_\alpha f_\alpha \overrightarrow{e_\alpha} + \frac{1}{2}\overrightarrow{F}\delta_t, \tag{9}$$

and the pressure field p may be written as

$$p = c_s^2 \rho. \tag{10}$$

The physics behind the kinetic equation Eq. (1), and in particular, the collision matrix $\Lambda_{\alpha\beta}$ will become more transparent when it is specified directly in terms of a set of linearly independent moments $\widehat{\mathbf{f}}$ instead of the distribution functions $\mathbf{f}$, i.e. through $\widehat{\mathbf{f}} = \left[\widehat{f}_0, \widehat{f}_1, \widehat{f}_2, \ldots, \widehat{f}_{18}\right]^t$. The moments have direct physical import to the macroscopic quantities such as momentum and viscous stress tensor (The components of $\widehat{\mathbf{f}}$ are provided in Appendix B). This is achieved through a transformation matrix $\mathcal{T}$:$\widehat{\mathbf{f}} = \mathcal{T}\mathbf{f}$ (d'Humi'eres et al., 2002). The elements of $\mathcal{T}$ are given in Appendix C. Each row of this matrix is orthogonal to every other one. The essential principle for its construction is based on the observation that the collision matrix Λ becomes a diagonal matrix $\widehat{\Lambda}$ through $\widehat{\Lambda} = \mathcal{T}\Lambda\mathcal{T}^{-1}$ in a suitable orthogonal basis which can be obtained as combinations of monomials of the Cartesian components of the particle velocity directions $\overrightarrow{e_\alpha}$ through the standard Gram-Schmidt procedure. As a result, this is a refinement over the earlier relaxation approaches through collision matrices (Higuera 1989a, 1989b) in that the relaxation is directly specified for the moments, in the spirit of the moment method by Grad (1949). The transformation matrix $\mathcal{T}$ is maintained row-wise orthogonal instead of orthonormal to reduce the floating-point operations involved in normalization. Thus, the collision matrix in moment space is greatly simplified and the description of the dynamics of fluid flow can be naturally related to the underlying physical model, which can be characterized through relaxation process of various fields based on kinetic theory (Grad, 1949).

The collision matrix in moment space may thus be written as

$$\widehat{\Lambda} = diag\left(s_0, s_1, s_2, \ldots, s_{18}\right), \tag{11}$$

where $s_0, s_1, s_2, \ldots, s_{18}$ relaxation time rates for the respective moments. The corresponding components of the local equilibrium distributions in moment space $\widehat{\mathbf{f}}^{eq} = \left[\widehat{f}_0^{eq}, \widehat{f}_1^{eq}, \widehat{f}_2^{eq}, \ldots, \widehat{f}_{18}^{eq}\right]^t$ are functions of the local density and momentum fields and is given in Appendix D. When there is an external force field, as in Eq. (4) represented in particle velocity space, appropriate source terms in moment space $\widehat{\mathbf{S}}$ need to be introduced. In an earlier work, the functional expressions of $\widehat{\mathbf{S}}$ in terms of force and velocity fields for the three-dimensional, fifteen velocity (D3Q15) particle velocities were derived (Premnath and Abraham, 2006). In this report, we summarize the final expressions for $\widehat{\mathbf{S}}$ for the D3Q19 model in Appendix E. Effectively, due to collisions and the presence of external forces, the distribution functions in moment space, or simply the moments, are modified by the quantity $-\Lambda\left(\widehat{\mathbf{f}} - \widehat{\mathbf{f}}^{eq}\right) + \left(I - 1/2\widehat{\Lambda}\right)\widehat{\mathbf{S}}\delta_t$

A multiscale analysis based on the Chapman-Enskog expansions (Chapman and Cowling, 1964) shows that in the continuum limit the GLBE corresponds to the weakly compressible Navier-Stokes equations with the density, velocity and pressure. The macrodynamical equations can also be derived through an asymptotic analysis under a diffusive scaling (Sone, 1990; Junk, 2001; Junk et al., 2005). The transport properties of the fluid flow, such as bulk ζ and shear ν kinematic viscosities can be related to the appropriate relaxation times through either Chapman-Enskog analysis of the GLBE or the von Neumann stability analysis of its linearized version (Lallemand and Luo, 2000):

$$\zeta = \frac{2}{9}\left(\frac{1}{s_2} - \frac{1}{2}\right)\delta_t, \tag{12}$$

$$\nu = \frac{1}{3}\left(\frac{1}{s_\beta} - \frac{1}{2}\right)\delta_t, \quad \beta = 9, 11, 13, 14, 15. \tag{13}$$

Notice that from Eq. (13) $s_9 = s_{11} = s_{13} = s_{14} = s_{15}$ to maintain isotropy of the stress tensor and s_2 determines the magnitude of bulk viscosity. The rest of the relaxation parameters do not affect hydrodynamics but can be chosen in such a way to enhance numerical stability to simulate higher Reynolds number problems for a given grid resolution, in particular for the bounded turbulent flow considered here. Based on linear stability analysis (Lallemand and Luo, 2000), the following values for the other relaxation parameters are determined (d'Humières 2002): $s_1 = 1.19$, $s_2 = s_{10} = s_{12} = 1.4$, $s_4 = s_6 = s_8 = 1.2$ and $s_{16} = s_{17} = s_{18} = 1.98$. For the conserved moments, the values of the relaxation parameters are immaterial as their corresponding equilibrium distribution is set to the value of the respective moments itself. However, with forcing terms it is important that they be non-zero (McCracken and Abraham, 2005; Premnath and Abraham, 2006). For simplicity, we set $s_0 = s_3 = s_5 = s_7 = 1.0$. It may be noted that all relaxation parameters have the following bound $0 < s_\alpha < 2$. In this paper, we employ the above values for the relaxation parameters. It may be noted that when all the relaxation times are equal, i.e., $s_1 = s_2 = \ldots = s_{18} = 1/\tau$, where τ is the relaxation time, the MRT reduces to the single-relaxation time (SRT) model. The SRT, first proposed to model collision process in the continuous Boltzmann equation by Bhatnagar, Gross and Krook (1954) was introduced in the LBE in Qian et al. (1992) and Chen et al. (1992) . This form of LBE is also referred to as the BGK-LBE. Owing to its apparent simplicity, it is in more common use. However, in particular for turbulence simulations with a subgrid scale (SGS) model, as will be shown later, the MRT has marked advantages when compared with the SRT: for a given resolution, MRT is more accurate and much more numerically stable with a insignificant additional computational overhead, thereby allowing higher Reynolds numbers to be reached than possible with the SRT.

On the subject of computational efficiency, it is important to note that the implementation of Eq. (1) does not involve direct summation as shown, but through a highly optimised procedure that exploits certain properties of $\lambda_{\alpha\beta}$ (see Appendix F).

3.2 Boundary conditions

For solid wall boundaries, the simplest boundary condition is the bounceback scheme. In this method, the populations are simply reflected:

$$f_{\overline{\alpha}}(\vec{x}, t + \delta_t) = \tilde{f}_\alpha(\vec{x}, t) \tag{14}$$

where $\overline{\alpha}$ is the direction opposite to α. In this equation, $\tilde{f}_\alpha$ is the distribution function after the collision step (and before the streaming step). This scheme is suitable for flat walls where the direction is parallel to a coordinate axis plane. In this situation, the wall is located half way between two nodes, and for this reason it is sometimes known as the halfway bounceback scheme.

A number of different options are available for curved boundaries. Those tested in this work were those by Mei et al. (1999), Guo et al. (2002), Bouzidi et al. (2002) and Lallemand and Luo (2003a). All these methods are based on applying interpolations/extrapolations around position of the boundary and then executing bounce-back type conditions at the boundary location. After assessment, we chose the so-called interpolated bounce back scheme (Bouzidi et al., 2001) due to its robustness. It specifies the distribution functions for the incoming particle directions from the wall in terms of a parameter q, where q represents the fractional distance of the wall from the near-wall lattice node in comparison with the lattice spacing (see Fig. 3). The first-order implementation of the scheme is:

$$f_{\bar{\alpha}}(\boldsymbol{x}, t + \delta t) = 2q\tilde{f}_\alpha(\boldsymbol{x}, t) + (1 - 2q)\tilde{f}_{\bar{\alpha}}(\boldsymbol{x} - \boldsymbol{e}_\alpha \delta t, t) \qquad \text{for} \quad q < 0.5 \tag{15}$$

$$f_{\bar{\alpha}}(\boldsymbol{x}, t + \delta t) = \frac{1}{2q}\tilde{f}_\alpha(\boldsymbol{x}, t) + \frac{2q - 1}{2q} f_{\bar{\alpha}}(\boldsymbol{x}, t) \qquad \text{for} \quad q \geq 0.5 \tag{16}$$

$$\tag{17}$$

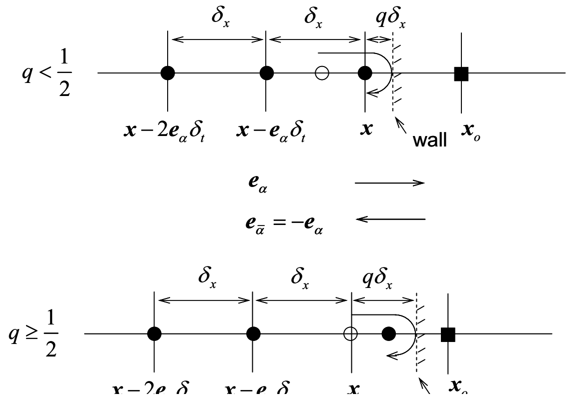
and the second-order version is:

$$f_{\bar{\alpha}}(\boldsymbol{x}_f, t + \delta t) = \frac{1}{q(2q+1)} \tilde{f}_{\alpha}(\boldsymbol{x}_f, t) \qquad (18)$$

$$+(1-2q)(1+2q)\tilde{f}_{\bar{\alpha}}(\boldsymbol{x} - \boldsymbol{e}_{\alpha}\delta t, t) + q(1-2q)\tilde{f}_{\alpha}(\boldsymbol{x} - 2\boldsymbol{e}_{\alpha}\delta t, t) \quad \text{for} \quad q < 0.5(19)$$

$$f_{\bar{\alpha}}(\boldsymbol{x}_f, t + \delta t) = \frac{1}{q(2q+1)} \tilde{f}_{\alpha}(\boldsymbol{x}, t) + \frac{2q-1}{q} \tilde{f}_{\bar{\alpha}}(\boldsymbol{x}, t) + \frac{1-2q}{1+2q} f_{\bar{\alpha}}(\boldsymbol{x} - \boldsymbol{e}_{\alpha}\delta t, t) \quad \text{for} \quad q \geq 0.5(20)$$

In practice, very little improvement was found by using the second order scheme over the first order, though both schemes gave substantially better results than just using the bounceback (to which the above reduces for $q = 0.5$) in which curved boundaries are resolved in "staircase" fashion. For inflow boundary surface with a specified velocity u_{in} , extended bounce back (Ladd, 1994) that adds appropriate momentum to the particle populations is implemented (see Fig. 4):



δ_x
δ_x
$q\delta_x$
$q < \frac{1}{2}$
$\boldsymbol{x} - 2\boldsymbol{e}_\alpha\delta_t$
$\boldsymbol{x} - \boldsymbol{e}_\alpha\delta_t$
$\boldsymbol{x}$
wall
$\boldsymbol{x}_o$
$\boldsymbol{e}_\alpha$
$\boldsymbol{e}_{\bar{\alpha}} = -\boldsymbol{e}_\alpha$
δ_x
δ_x
$q\delta_x$
$q \geq \frac{1}{2}$
$\boldsymbol{x} - 2\boldsymbol{e}_\alpha\delta_t$
$\boldsymbol{x} - \boldsymbol{e}_\alpha\delta_t$
$\boldsymbol{x}$
$\boldsymbol{x}_o$

Figure 3: Wall boundary implementation for cut-lattice cells.

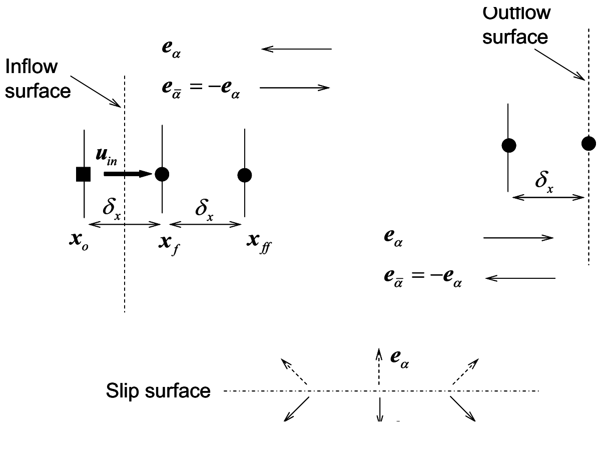


Figure 4: Implementation of inflow, outflow, and symmetry boundary conditions.

$$f_{\overline{\alpha}}(\vec{x}, t + \delta_t) = \tilde{f}_{\alpha}(\vec{x}, t) + 2w_{\alpha}\rho_{in}\frac{\boldsymbol{e}_{\overline{\alpha}} \cdot u_{in}}{c_s^2} \qquad (21)$$

For outflow boundaries, an extrapolation method (Chen et al., 1996) is employed

$$f_{\overline{\alpha}}(\vec{x}, t + \delta_t) = 2\tilde{f}_{\overline{\alpha}}(\vec{x}, t) - \tilde{f}_{\overline{\alpha}}(\vec{x} - \boldsymbol{e}_{\alpha}\delta_t, t) + 2w_{\alpha}\rho_{in}\frac{\boldsymbol{e}_{\overline{\alpha}} \cdot u_{in}}{c_s^2} \qquad (22)$$

in which the equilibrium distribution in the collision step for the boundary lattice nodes are specified in terms of the no-gradient conditions for the macroscopic quantities. For a symmetry surface (see Fig. 3), we use the condition $f_\beta(\boldsymbol{x}, t+\delta_t) = \tilde{f}_\alpha(\vec{x}, t)$. After implementation of these methods, initial computations were carried out for a set of cases involving laminar flows to assess the suitability. These tests yielded results that were in excellent comparison with analytical solution and other computational solutions, demonstrating the suitability of these approaches.

3.3 Subgrid scale (SGS) turbulence modelling

We have implemented the standard Smagorinsky model to perform large eddy simulations (LES). The eddy viscosity ν_t for this model can be written as

$$\nu_t = (C_s \Delta)^2 \overline{S}, \quad \overline{S} = \sqrt{2S_{ij}S_{ij}} \tag{23}$$

where C_s is a constant (this was set to 0.12 for the work described in this report). Here, Δ is the cut-off length scale set equal to the lattice-grid spacing, i.e. $\Delta = \delta_x$, and S_{ij} is the strain rate tensor given by $S_{ij} = 1/2\left(\partial_j u_i + \partial_i u_j\right)$. In LBM, the strain rate tensor can be computed directly from the non-equilibrium part of the moments, without the use of finite-differencing of the velocity field. Recently, Luo derived the expressions for the strain-rate tensor in terms of moments for the D3Q19 model of GLBE without forcing term in Yu et al. (2006). In this report, we extend the results for GLBE with forcing term and the final results are summarized in Appendix F. This procedure for calculation of strain rates in the MRT model is fully local in space and is computationally efficient, particularly for complex geometries. The eddy viscosity ν_t is added to the molecular viscosity ν_0, obtained from the statement of the problem, through the characteristic dimensionless number, such as shear Reynolds number for turbulent channel flow problem, to yield the total viscosity ν (i.e., $\nu = \nu_0 + \nu_t$). Subsequently, the relaxation times may be obtained from Eq. (13). When such eddy-viscosity type SGS models are used to provide additional contributions to the relaxation times, the MRT model can be considered to be "coarse-grained" and it can be readily shown that the macroscopic dynamical equations of fluid flow corresponds to the filtered equations with the SGS Reynolds stress represented through the eddy viscosity. The distribution functions, or equivalently, the moments, including the hydrodynamic fields, can be considered to be grid-filtered quantities. An alternative approach is to directly apply filters to the moment representation of the GLBE and rigorously "derive" SGS models essentially from kinetic theory under appropriate scaling (Ansumali et al., 2004);

To account for the damping of scales near the walls, following an earlier work (Moin and Kim, 1982), we have implemented the van Driest damping function

$$\Delta = \delta_x \left[1 - \exp\left(-\frac{z^+}{A^+}\right)\right] \tag{24}$$

where $z^+ = zu_*/\nu_0$ is the normal distance in wall units from the wall, where u_* is the friction velocity related to the wall shear stress τ_w through $u_* = \sqrt{\tau_w/\rho}$, and A^+ is taken equal to 25. While this approach has some empiricism built-in, for simple wall-bounded turbulent flows, such as the problem considered here it has been shown to be reasonably accurate in prior work based on the solution of grid-filtered Navier-Stokes equations. Also, as will be shown later in this report that the MRT model is able to reproduce turbulence statistics in the near-wall region reasonable well using this damping supplemented to the SGS model. A more advanced approach is to use dynamic SGS models (e.g., Germano et al., 1991; Zhang et al., 1995; Salvetti and Banerjee, 1995) for LES, and this is a subject for future work.

3.4 Accelerating convergence to steady-state

The equations so far presented have been for the general case of transient flows. For steady state cases, it is possible to reformulate the equations such that they can be solved more quickly with computational methods. This can be done by introducing a pre-conditioning factor γ into the Navier–Stokes equations:

$$\frac{\partial}{\partial t}(\rho\boldsymbol{u}) + \frac{1}{\gamma}\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u}) = \frac{1}{\gamma}\boldsymbol{\nabla}p + \frac{1}{\gamma}\boldsymbol{\nabla}\cdot(\rho\nu\boldsymbol{\nabla}\boldsymbol{u}) + \frac{1}{\gamma}\boldsymbol{F} \tag{25}$$

where $\boldsymbol{F}$ is a body force that may include contributions from external forces such as buoyancy or MHD effects. A suitable preconditioned MRT model is developed that recovers Eq. (25) (Premnath and Pattison, 2007) By adjusting the parameter γ, the speed at which convergence to steady state can be achieved can be considerably reduced. The lower the value of γ, the faster the speed up – a value of $\gamma = 1$ corresponds to the transient solution with no acceleration. The minimum value of γ that can be used is determined by the maximum fluid velocity – for flows with low mach numbers, much lower values can be used without stability problems creeping in. MetaFlow automatically computes a suitable value of γ. The details of the implementation are given in Appendix H.

Figure 5 shows how the value of γ influences the rate at which convergence of the computation takes place. The case chosen was a simple pressure-driven flow, plane Poiseuille flow and results are shown for two different viscosities. These show a considerable speed up. Note that the viscosities have been normalised with the grid spacing and time step, as will be the case for most the plots discussed in this report.

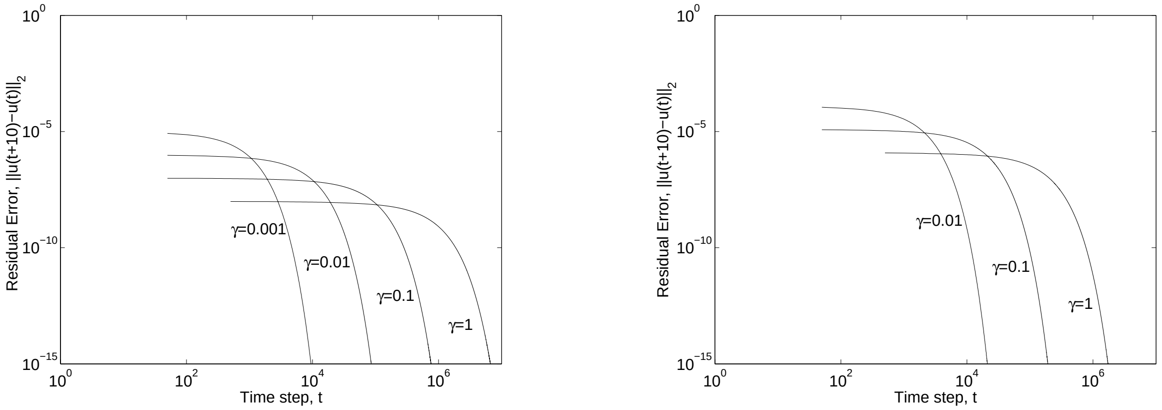


Figure 5: Effect of γ on the rate of convergence of plane Poiseuille flow.

A thorough investigation of the preconditioning was undertaken as part of the project, and a detailed discussion is presented in the paper by Premnath and Pattison (2007).

It should be noted that the preconditioning approach taken here was based on work in a paper by Guo et al. (2004); however it was noticed that their method was limited to cases with constant body forces (no spatial variation), and this shortcoming is corrected here. Moreover, their approach is limited to the SRT model which has been generalised by Premnath et al. (2007) for the MRT model.

3.5 Parallelisation and performance on a multiprocessor cluster

A key advantage of using the LBM over finite difference/finite volume (FD) methods is that they are very amenable to parallelisation. MetaFlow has been parallelised using the Message Passing Interface (MPI) technique for distributed memory machines. The key step involved in parallelisation is to divide the computational domain into a number of subdomains, each of which is assigned to a different processor. After each time step, each processor exchanges data with a number of other processors. In the LBM, computations only require information from nearby adjacent points (and in some cases from points two nodes away). What this means is that a processor only needs to exchange data with the processors dealing with adjacent sub-domains; in most cases this means they need to transfer data to two or perhaps eight other processors. This contrasts with other schemes where each processor may have to receive information from every other processor. This can place a limit on how many processors can usefully be employed.

A series of tests were performed by running a simple channel flow case with a Smagorinsky SGS turbulence model on the NCSA machine at Urbana–Champaign. It was found that as long as the width of the subdomains remained reasonable large (above about 12 points) the proportion of time spent communicating stayed low, as would be expected. Figure 6 shows the speed of the computer simulation as a function of the number of processors used. A factor of about 14 was achieved with 16 processors, which should be considered very good – if the number of points in the streamwise direction had been a multiple of 16 (to ensure equal distribution among the processors), even better performance would be expected.

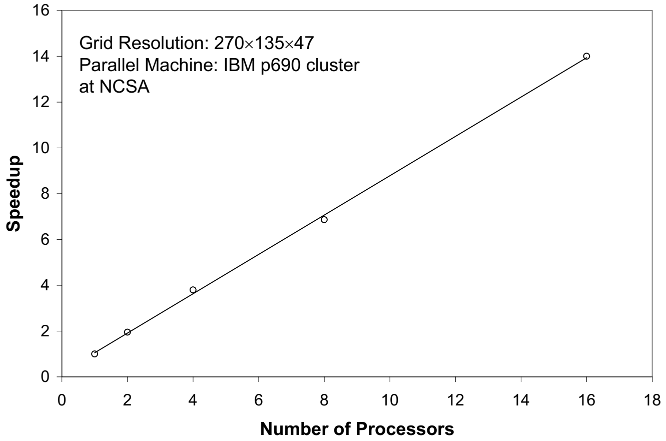


Figure 6: Variation of computing speed with number of processors.

Another test was to compare the speed of our LBM with that of a finite difference code. The code selected for comparison was an established LES solver which used a second order finite difference scheme (Salvetti and Banerjee, 1995; Salvetti et al., 1997). The simulations were set to run the same number of grid points for the same problem, channel flow with a Smagorinsky turbulence model. Both the codes were run on a single processor of AMD Athlon 2000 dual processor computer running the Linux operating system. The GNU compiler with the -O3 optimisation was used and checks were made to ensure that the results were reproducible – it was found that the times taken did not vary significantly (within about 1%).

Grid Size	LB code	Finite Difference	Ratio
270 × 135 × 47	516	2015	3.9
180 × 90 × 32	153	505	3.3
135 × 135 × 92	510	1824	3.6
135 × 270 × 47	516	1849	3.6
90 × 45 × 32	38	88	2.3

Table 2: Computational times of MRT and FD codes running LES of a turbulent channel on a single processor.

Table 2 shows the times required for 100 time steps for each code. It shows that as far as the computational cost per grid node and per time step is concerned, the MRT code runs faster than the FD code employed. Despite being substantially more complex, the MRT only about 30% slower than the SRT, probably because the nature of the MRT permits many optimisations to be performed. The limiting time step was also examined, for the MRT it was found to be at a CFL number of about 0.42, whereas the FD code became unstable at about 0.35. However the accuracy at the largest time steps may not be so good. In any case, GLBE implementation is competitive as compared to the FD code. An important point to note here is the elimination of the need to solve the time consuming Poisson-type equation for pressure field by the GLBE, but which is inherent in most of the FD methods. It is also interesting to observe that a very recent work based on a LBE for DNS of turbulence flows reported it to be competitive in terms of run-times when compared with the Chebyshev-spectral methods on single processors (Lammers et al., 2006).

After the MRT model had been implemented in our code, it soon became clear that it was much more stable than the earlier SRT model. Simulations of some of the more challenging cases suggested that it was more stable by a factor of about three, a value in keeping with previous work. Also, for the simple case of turbulent channel flow, it did not produce the spurious spikes in turbulent fluctuations near the wall which had been observed with the SRT implementation on relatively coarser grids. To obtain a more quantitative assessment of the stability, tests were made with a lid-driven cavity flow. Following the example of d'Humières et al. (2002), the lid velocity was imposed by setting the distribution functions at the lid to the equilibrium distribution function. This method is more stable than alternative methods for setting the velocity, but it should be noted that it is unsuitable for setting velocities where there is a component normal to the wall, for which the Ladd (1994) scheme discussed earlier is recommended. In these tests, the velocity of the lid was gradually increased until the computation failed (i.e. crashed or showed unphysical behaviour). If the velocity was increased slowly enough, the point at which this occurred was found to be reproducible. Table 3 shows the velocities at which the computation became unstable – it is clear that the MRT performed much better than the SRT on this test. Furthermore, it should be noted that the transition to turbulence takes place at a Reynolds number of about 12 000, so the SRT was unable to reach the turbulent region on any of the runs, whereas the MRT was, when similar grid resolutions are maintained.

Grid	Viscosity	Maximum velocity		Maximum Re		Ratio
64^3	0.0005	0.0295	0.149	3776	19100	5.1
64^3	0.001	0.059	0.232	3776	14848	3.9
64^3	0.002	0.123	0.385	3936	12320	3.1
96^3	0.0005	0.0265	0.188	5088	36096	7.1

Table 3: Point at which LBM became unstable for lid driven cavity flow.

3.6 Validation examples

As part of the development of the LBM code, numerous validation studies have been performed. Some examples with turbulent flow are briefly discussed in this section. More thorough analyses are being submitted to journals as papers.

3.6.1 Turbulent channel flow

The channel flow case involved the simulation of a free surface flow with a shear Reynolds number $Re_* = u_* H/\nu$ where H is the channel depth, ν is the molecular viscosity, and u_* the friction velocity. The case is documented fully in Premnath et al. (2007), but the main results are presented here too.

A schematic diagram of the channel flow problem with the top surface being considered as slip surface is shown in Fig. 7.

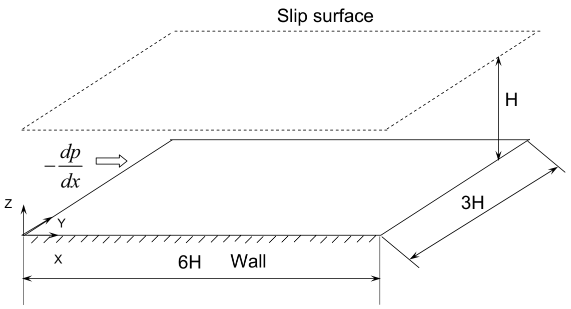


Figure 7: Schematic diagram of channel flow simulation.

As shown in this figure, the computational domain is chosen with appropriate aspect ratios to sample wall-layer streaks adequately and to let proper decay of two-point correlations (Kim, Moin and Moser, 1987). We considered a uniform grid, and chose the height H to be resolved by 45 grid nodes such that spacing in wall units becomes 4.08. The first grid point inside the fluid is half this distance from the wall, 2.08 wall units. The computational domain thus consists of 270×135×45 grid points.

The initial mean velocity is specified to satisfy the 1/7th power law with initial perturbations superimposed to initiate turbulence. The perturbations are specified such that they satisfy divergence free velocity field (Lam, 1989; Lam and Banerjee, 1992). The flow is driven by pressure gradient in the x or streamwise direction. The friction velocity can be related to the gradient by $dP/dx = \rho u_*^2/H$. The computation was run until a dynamically steady state was achieved, then continued for a sufficiently long time to collect statistics.

Figure 8 shows the computed mean velocity profile as a function of the distance from the wall and compares it with predictions for the viscous layer and log law regions. The axes are normalised with wall units (ν/u_*) and friction velocity. It can be seen that the computations agree reasonably well with the viscous sub layer and log-law wall scalings, there is a slight overprediction, but this is generally expected with LES.

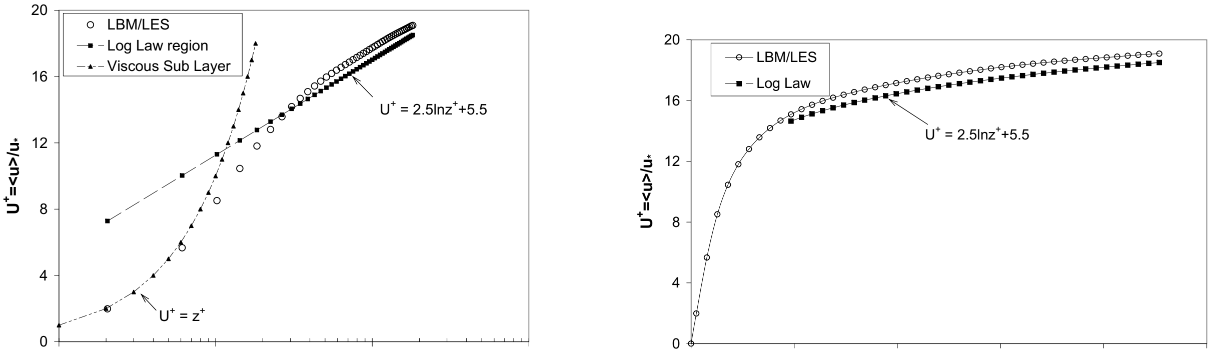


Figure 8: Comparison of the computed mean velocity profile with scaling laws for fully-developed turbulent channel flow at $Re_* = 183.6$.

Figure 9 shows the comparisons of the components of the root-mean-square (rms) streamwise, spanwise and wall-normal velocity fluctuations, respectively, computed using LBM with DNS (Kim, Moin and Moser, 1987) and experimental (Kreplin and Eckelmann, 1979) data. Computed results agree reasonably well with prior data.

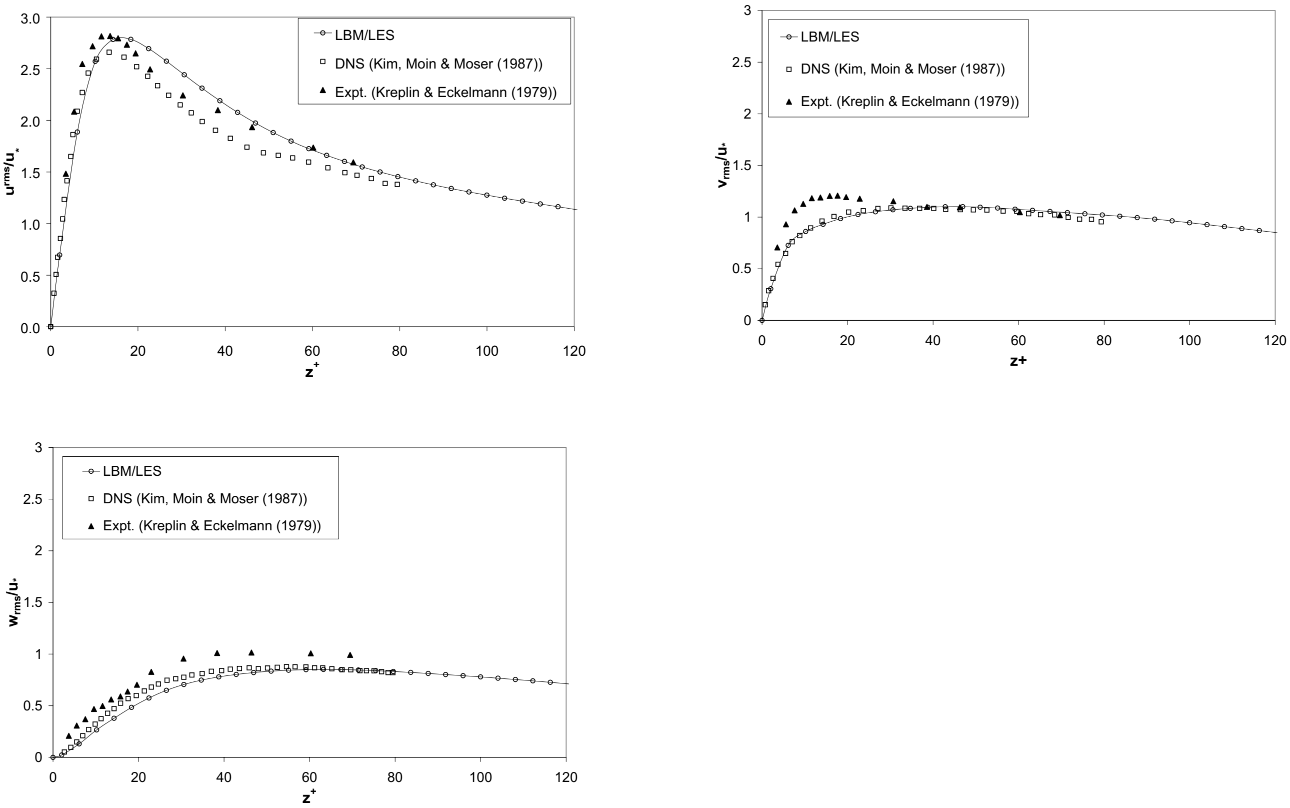


Figure 9: Root-mean-square streamwise velocity fluctuations for fully developed turbulent channel flow at $Re_* = 183.6$.

3.6.2 Turbulent flow through a square duct

Turbulent fluid flow through a square duct is characterised by the existence of net flows in directions perpendicular to the bulk flow. These circulations, frequently termed *secondary flows of the second kind*, also arise in turbulent flow through channels of other non-circular cross-sections, but are not found in laminar flows. Although long recognised to be associated with the turbulence, the exact cause of these circulations has been a subject of debate, though it is generally recognised to be associated with gradients in the Reynolds stress in the cross-sectional plane. A more detailed explanation is given by Huser and Birigen (1993) who describe the mechanisms in terms of the interaction of turbulence structures generated at the wall.

In a square duct, these secondary flows manifest themselves as a set of eight vortices, each one enclosed by a wall, a corner bisector and a wall bisector. The general features are not sensitive to the Reynolds number, except that it should be high enough that the flow be fully turbulent. The velocities associated with these flows is relatively small, of the order of 1% of the mean streamwise velocity, and thus significantly smaller than the turbulent velocity fluctuations. However, since they do lead to net flows in directions perpendicular to the walls, they can have a significant effect on heat transfer and passive scalar transport. Despite their weak nature, these secondary circulations are therefore of great interest to a number of engineering problems.

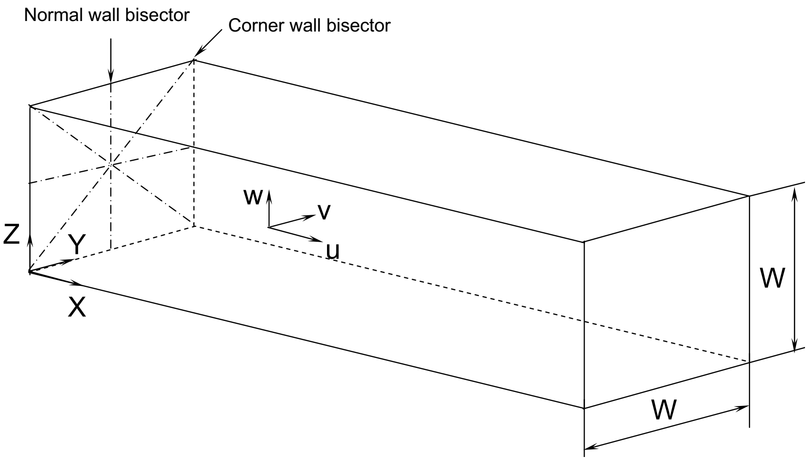


Figure 10: Schematic diagram showing geometry modelled.

The prediction of these secondary flows is relatively difficult. For example, the ability of RANS to capture the secondary circulations is dependent on the relations used for the stresses, for example a simple k-ϵ model will not reveal these motions. Since the coefficients in Reynolds stress models are problem-specific and often not well known, they are not well suited for predicting phenomena which are particularly sensitive to parameter choices.

This was chosen as a validation case for the LBM as the prediction of these secondary flows is quite challenging, and would be a good test of whether our model could properly capture the turbulent behaviour. DNS data of Gavrilakis (1992) and Huser and Birigen (1993) were available for comparison, as was LES of Madabhushi and Vanka (1991). The full analysis can be found in Pattison et al. (2007), and a selection of the results is presented here.

A schematic representation of the geometry used is shown in Fig. 10. The Reynolds number used was 300. This is defined as $\mathrm{Re}_* = u_* w/\nu$ where w is the duct width and the mean friction velocity is calculated from the imposed pressure gradient as $\rho u_*^2 = w\,|dP/dx|/4$. The dimensionless viscosity (normalised by the grid spacing and time step) was set to 0.001.

Periodic boundary conditions were applied in the x (streamwise) direction, and no-slip conditions in the other two directions. The use of grid points within the wall meant that the boundary was half a grid spacing from the outermost points, thus the effective width was 72 grid separations. A uniform grid was used for the simulation, though it should be noted that stretched grids can be used with the LBM (Premnath and Abraham, 2004). The choice of domain proportions followed the recommendation of Huser and Birigen (1993), who found that a streamwise extent of about six times the duct width was sufficiently long to yield accurate simulations.

For the initial conditions, an approximate velocity field was set up, based on the 1/7 power law. As with the channel flow case, in order to initiate turbulence, a perturbation profile with sinusoidal fluctuations was superimposed on this, chosen so as to satisfy continuity.

The simulation was run using the MRT model with a pressure gradient providing the driving force. A single processor Pentium IV machine was found to be sufficient for this. The computation was run until a dynamically steady state was reached, and then continued for a period of over 100 000 time steps over which statistics were collected.

Three main sources of data were used for comparisons with the MetaFlow simulations, and they are summarised in Table 1. The DNS of Gavrilakis was performed at the same Reynolds number as our simulation and these results are likely to be very accurate. Comparisons with Madabhushi and Vanka's data are included to enable the performance of the LBM LES code to be judged against that of a pseudo–spectral LES code.

Figure 11a shows contours of the streamwise velocity in the cross-sectional plane. In this plot, the velocity has been averaged over both time and the length of the domain. Figure 11b shows the secondary velocities (the non-streamwise component); in this case, and in all other plots unless otherwise stated, statistics were also averaged over the eight similar octants.

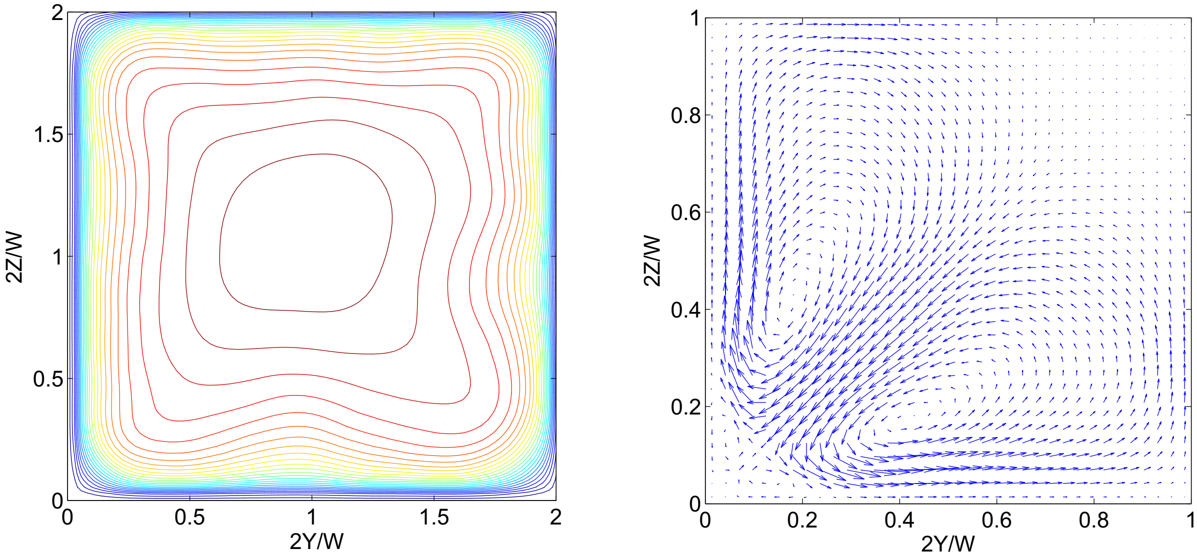


Figure 11: Mean velocities in the cross-sectional plane: (a) streamwise velocity contours (b) secondary velocity vectors.

Secondary flow vectors Figure 11b clearly shows the expected vortex moving in a sense such that fluid moves toward the corner along the corner bisector and away from the wall along the wall bisector. The vortex is centred at about (0.45, 0.18) where the numbers in parenthesis are the y and z coordinates normalised with respect to the half width. This compares well with Gavrilakis's DNS which gave the centre at (0.5, 0.2). Madabhushi and Vanka's LES predicted the centre at (0.55, 0.25) and Huser and Birigen's DNS, (at higher Reynolds number) at (0.4, 0.2). The plot also suggests the presence of a small vortex very close to the corner – this was also noticed by Gavrilakis.

The contours of streamwise velocity can be seen to bend toward the walls near to the corners. This is a characteristic of duct flow and is associated with the secondary flow. The secondary vortex transports faster–moving fluid from the central region toward the corners along the corner bisectors while slower–moving fluid from the vicinity of the wall is advected toward the centre near the wall bisectors, resulting in the bulge in the contours.

Figure 12 shows the mean streamwise velocity along the wall bisector. To be consistent with the data with which it is compared, the velocity is normalised using the local friction velocity at the wall bisector, calculated as $u_\tau^2 = \tau/\rho$ where τ is the wall stress and ρ the density. Distance is plotted in terms of wall units ν/u_τ, where ν is the viscosity. When compared with Gavrilakis's DNS, the mean velocity is slightly overpredicted, as is often the case with the lower resolutions used with LES. Huser and Birigen used a lower resolution in the wall-normal direction than Gavrilakis and the is the likely reason for the higher velocity predicted, though there could also be Reynolds number effects. Also, Huser and Birigen had performed a preliminary DNS at a coarser resolution, and this had given higher velocities than the

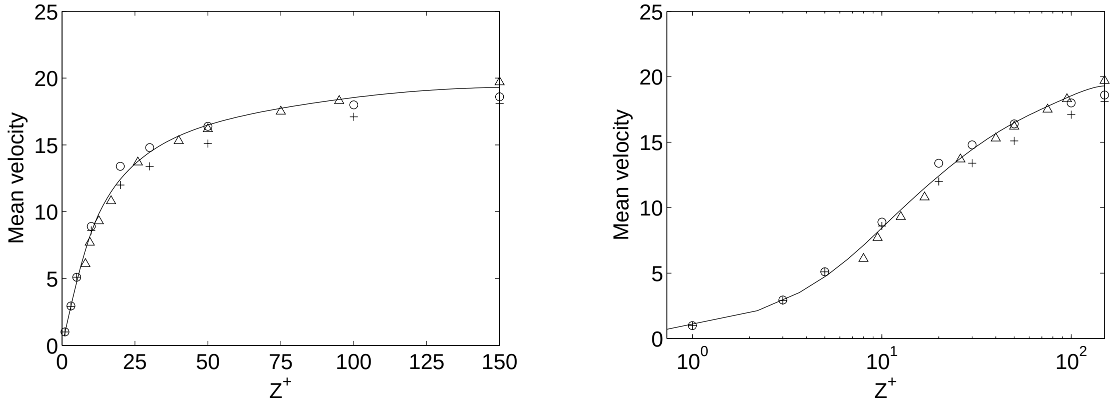


Figure 12: Mean streamwise velocity along wall bisector. Lines are MetaFlow predictions, circles Huser & Birigen, triangles Madabhushi & Vanka, and crosses Gavralakis.

final DNS.

Figure 13 shows the secondary velocity plotted along a corner bisector, normalised with the mean friction velocity, u_*. The results are compared with those of Gavrilakis, and good agreement can be seen. The secondary velocity exhibits a minimum close to the corner, which is indicative of a small secondary vortex in this region. Gavrilakis also observed such a vortex in the region close ($y < 0.1w/2$) to the corner, but did not provide any quantitative values for the secondary velocity in this region. Profiles for the spanwise (y) component of velocity are plotted as a function of z for fixed values of y in Fig. 14. Comparisons are made with both DNS results Gavrilakis (1992) and experimental data (Cheesewright et al., 1990) taken at a similar, though slightly lower, Reynolds number.

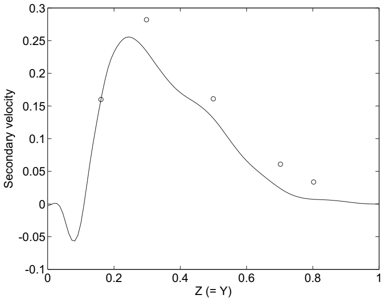


Figure 13: Magnitude of secondary flows $v\sqrt{2} = w\sqrt{2}$ along a corner bisector. Circles are DNS data from Gavrilakis.

Turbulence statistics are presented in Fig. 15 which shows the root mean square (rms) velocity fluctuations along a wall bisector, together with data from LES and DNS for comparison. The level of agreement with the DNS data can be seen to be good, and MetaFlow shows significantly better agreement with the DNS data than the other LES, despite the fact that MetaFlow was run with a much coarser resolution in the important near-wall region. The variation in rms velocities is very similar to that found in other wall–bounded flows and comparisons with plane channel data can be found in Gavrilakis's paper. Gavrilikas raised the question of whether the friction velocity at the wall centre, u_τ, or the average friction velocity, u_*, should be used for the normalisation of the data, and noted that agreement with channel data was best if u_τ was used near to the wall and u_* farther out.

In conclusion, the MRT model has demonstrated its ability to simulate turbulence in these test cases, and its ability to capture the secondary flows in the square duct is particularly encouraging.

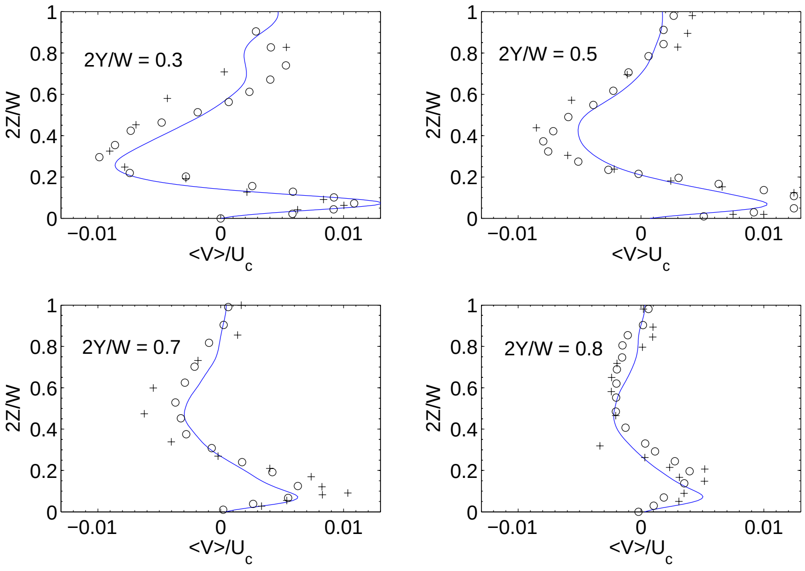


Figure 14: Magnitude of secondary flows $v\sqrt{2} = w\sqrt{2}$ along lines of constant y. Circles are DNS data from Gavrilakis and crosses experimental data (Cheesewright et al.).

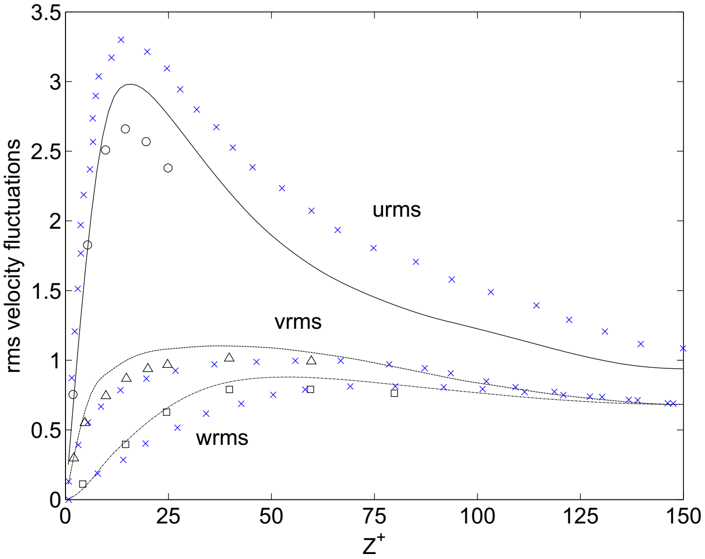


Figure 15: Root mean square velocity fluctuations along wall bisector. Lines are MetaFlow predictions for Re_*=300, open symbols DNS data of Gavrilakis (1992), and crosses LES data from Madabushi & Vanka (1991). Velocities have been normalised with u_τ.

4 Lattice Boltzmann Model for Magnetic Induction Fields

4.1 Basic equations

Magnetohydrodynamic effects are of particular importance to many systems of interest to fusion researchers, since liquid metals are commonly used. In these flows, the equations of motion take the same form as in non-MHD flows, but the body force term includes a component due to the Lorentz force arising from the interaction of induced current with magnetic induction. The hydrodynamic equations are:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0 \tag{26}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = \boldsymbol{\nabla} P + \boldsymbol{F} + \boldsymbol{\nabla} \cdot (2\nu\rho \boldsymbol{S}) \tag{27}$$

where the contribution to the body force term $\boldsymbol{F}$ from the Lorentz force is given by:

$$\boldsymbol{F}_{Lorentz} = \boldsymbol{J} \times \boldsymbol{B} = \boldsymbol{\nabla} \cdot \left[-\frac{1}{2\mu} \boldsymbol{B} \cdot \boldsymbol{B}\,\mathrm{I} + \frac{1}{\mu} \boldsymbol{B} \cdot \boldsymbol{\nabla} \boldsymbol{B} \right] \tag{28}$$

The magnetic field, $\boldsymbol{B}$, can be calculated from the magnetic induction equation. The form normally used for this is:

$$\frac{\partial \boldsymbol{B}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{u}) = \eta \nabla^2 \boldsymbol{B} \tag{29}$$

where η is the magnetic resistivity or diffusivity, defined by $\eta = 1/\sigma\mu$ and σ and μ are the electrical conductivity the magnetic permeability respectively. Note that the above equation assumes constant η; in practical situations, liquid metals may flow through channels with insulating walls, rendering this assumption invalid. A rigorous derivation of the equations from Maxwell's equations and Ohm's law was carried out to identify what extra terms may need to be included when η is a function of spatial coordinates, and this is presented in Appendix K. The extra terms do not appear to have been included in most other MHD codes, but may be incorporated in the next version of MetaFlow and the effect is assessed.

Using the LBM, the equation for the magnetic field can be solved in a similar fashion to the fluid dynamic equations. A solution procedure based on that of Dellar (2002) was incorporated into MetaFlow; in this model, a vector distribution function $\mathbf{h}$ is employed, and the following equation is used for its evolution:

$$\tilde{\boldsymbol{h}}_i(\mathbf{x}, t) = \boldsymbol{h}_i(\mathbf{x}, t) - [\boldsymbol{h}_i(\mathbf{x}, t) - \boldsymbol{h}_i^{eq}(\mathbf{x}, t)]/\tau_m \tag{30}$$

$$\boldsymbol{h}_i(\mathbf{x} + \mathbf{e_i}, t + 1) = \tilde{\boldsymbol{h}}_i(\mathbf{x}, t) \tag{31}$$

The values with a tilde are known as post-collision values, as was the case with the hydrodynamic equations. The relaxation time τ_m which is a function of the conductivity and permeability of the medium.

$$\tau_m = \frac{1}{2} + 4\eta_m \frac{\delta t}{(\delta x)^2} \tag{32}$$

In practical implementations of this scheme, it is found that only seven directions are required; these are taken to be directions parallel to the axes and a zero vector, i.e. the first seven directions used in the lattice used for the fluid dynamic calculations.

The equilibrium distribution functions $\boldsymbol{h}_i^{eq}$ are calculated through:

$$h_{\alpha i}^{eq} = w_\alpha \left[B_i + \frac{4e_{\alpha j}^m}{c^2}(u_j B_i - B_j u_i) \right] \tag{33}$$

The magnetic field can be recovered from the model by taking zeroth kinetic moments:

$$B_i = \sum_\alpha h_{\alpha i} \tag{34}$$

The current can be calculated in two ways. One is of course from taking the curl of the magnetic field. The other, which avoids the need to taken finite differences, is to extract the current from the higher kinetic moments:

$$J_k = -\frac{4}{\mu_m c^2 \tau_m} \epsilon_{ijk} \sum_{\alpha=0}^{b_m} (e_{\alpha i}^m g_{\alpha k} - e_{\alpha i}^m g_{\alpha k}^{eq}) \tag{35}$$

where

$$\sum_{\alpha=0}^{b_m} e_{\alpha i}^m g_{\alpha k}^{eq} = (u_j B_i - B_j u_i) \tag{36}$$

A number of validation cases have been used to assess the performance of MetaFlow for flows with MHD effects. These range from one dimensional Hartmann flow to three-dimensional lid-driven cavity with an imposed magnetic field. In this report, results for the most demanding case, the MHD lid-driven cavity, are shown – other validations have been presented in papers arising from this work (see Appendix A).

4.2 Boundary conditions

For the induction equation, no suitable formulations for the boundary conditions could be found in the literature. Dellar (2002) had used a *reverse bounceback scheme* for the boundaries, but this method was only suitable for a limited range of geometries and did not allow an induced component in the direction parallel to the imposed field.

A new method of applying boundary conditions for the applied field was therefore formulated. The approach taken was similar to the extrapolation scheme used for the fluid flow. This is an extrapolation method and is similar to that described by Chen et al. (1996), except applied to magnetic field rather than fluid fluxes. The procedure is:

- for nodes on domain boundary (e.g. point a in Fig. 16), calculate post-collision values, $\tilde{\boldsymbol{h}}_\alpha$ using the imposed field $\boldsymbol{B}_0$ in place of that calculated from the zeroth moment of the distribution functions, appearing through the equilibrium distribution (see Eq. 33).
- calculate the post-advection values, $\boldsymbol{h}_\alpha$, using the extrapolation scheme

$$\boldsymbol{h}_\alpha(\boldsymbol{x}, t+\delta t) = 2\tilde{\boldsymbol{h}}_\alpha(\boldsymbol{x}, t) - \tilde{\boldsymbol{h}}_\alpha(\boldsymbol{x} - \boldsymbol{e}_\alpha, t) \tag{37}$$

In practice, since the distribution function has seven direction vectors, only one of which points into the domain, only one distribution value for each face is calculated is calculated with the above. For reasons of stability it was found that the two points neighbouring a face on which the magnetic field should be imposed should have the same electrical conductivity. For example to impose a insulating wall boundary condition, in Fig. 16), the grid points a and b should both lie within the solid region (which of course is assigned a very low conductivity). The insulating boundary will then lie midway between points b and c. In practice, with this scheme that one more extra layer of grid points than might otherwise have been needed is required. Conducting walls can also be similarly modelled handled by selecting appropriate values of the conductivity for the wall.

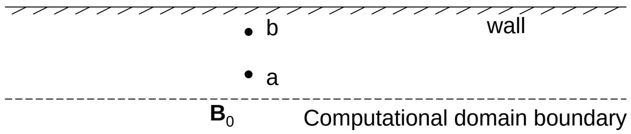


Figure 16: Boundary of flow in MHD case.

4.3 Application to steady-state MHD flows with low magnetic Prandtl numbers

One problem that had not previously been addressed by other researchers working on the LBM was that the timescales for the induction and the flow equations differ by several orders of magnitude for most practical situations. The magnetic Prandtl number is often used to describe this. This quantity is defined as the ratio of the viscosity to the resistivity:

$$\mathrm{Pr}_m = \frac{\nu}{\eta_m} \tag{38}$$

For a liquid metal, the Prandtl number may typically be of the order 10^{-7}, with the fluid dynamics being dominated by convection and the magnetic induction by diffusion. The relaxation time for the fluid flow go as the reciprocal of the viscosity and for the magnetic induction as the reciprocal of the resistivity. Appropriate treatment to handle this large disparity is required to ensure stability of the system. A method that overcomes this problem and allows "real" metals to be modelled was derived by introducing a scaling factor in a preconditioning approach that also allows faster convergence to steady state. Essentially this involves introducing a preconditioning factor γ_m and a scaling factor χ, such that the induction equation becomes:

$$\frac{\partial \boldsymbol{B}}{\partial t} + \frac{\chi}{\gamma_m}\boldsymbol{\nabla}\cdot(\boldsymbol{uB} - \boldsymbol{Bu}) = \frac{1}{\gamma_m}\eta\nabla^2\boldsymbol{B} \tag{39}$$

and the current must then be calculated with:

$$\boldsymbol{J} = \frac{1}{\chi\mu_m}\boldsymbol{\nabla}\times\boldsymbol{B} \tag{40}$$

In this above, χ is a scaling parameter to change the effective Prandtl number, $\mathrm{Pr}_{\text{eff}} = \chi\mathrm{Pr}$, and γ_m is a parameter that can be used to speed up the rate of convergence to a steady state. It should be noted that this approach is strictly only valid for steady state problems, which is often the case in fusion MHD problems, though in practice transients could be modelled provided the parameters are adjusted such that the magnetic field relaxes much more rapidly than the flow field. The relaxation parameter τ_m becomes

$$\tau_m = \frac{1}{2} + \frac{4\eta_m}{\gamma_m}\frac{\delta t}{\delta x^2} \tag{41}$$

Note that the computation will become unstable as τ_m increases too far above one and will lose accuracy. Similarly, it will become unstable as τ_m approaches 0.5. A full study of the effect of this scheme on the stability and convergence rates of the LBM is presented in one of our papers (Premnath and Pattison, 2007) but a few of the results are also given here. Appendix H provides more details, along with the equations used for the steady-solution hydrodynamics.

Other equations that need modification are the relation for the equilibrium distribution and the current

$$h_{\alpha i}^{eq} = w_\alpha \left[B_i + \frac{4e_{\alpha j}^m \chi}{c^2 \gamma_m} (u_j B_i - B_j u_i) \right] \tag{42}$$

$$J_k = -\frac{4}{\chi \mu_m c^2 \tau_m} \epsilon_{ijk} \sum_{\alpha=0}^{b_m} (e_{\alpha i}^m g_{\alpha k} - e_{\alpha i}^m g_{\alpha k}^{eq}) \tag{43}$$

where

$$\sum_{\alpha=0}^{b_m} e_{\alpha i}^m g_{\alpha k}^{eq} = \frac{\chi}{\gamma_m} (u_j B_i - B_j u_i) \tag{44}$$

One of the first tests was to check the model performance for Hartmann flow, the flow between two parallel plates with a magnetic field applied in the wall-normal direction. This flow, which is equivalent to Poiseuille flow with an imposed field, is characterised by the Hartmann number, Ha, given as

$$\mathrm{Ha} = B_0 L \sqrt{\frac{\sigma}{\rho\nu}} \tag{45}$$

where L is a length scale, in this case the half-width of the channel. Figure 17 shows the profiles of magnetic field and velocity along the wall-normal direction, and comparison with the analytical solution shows very close agreement. In these simulations, the magnetic resistivity was set equal to the viscosity and the effective Prandtl number is set to the desired value by varying the scaling factor χ.

$$\eta = \frac{\gamma_m}{4} \left(\tau_m - \frac{1}{2} \right) \frac{(\delta x)^2}{\delta t} \tag{46}$$

where γ_m can be adjusted such that it accelarates convergence while maintaining numerical stability.

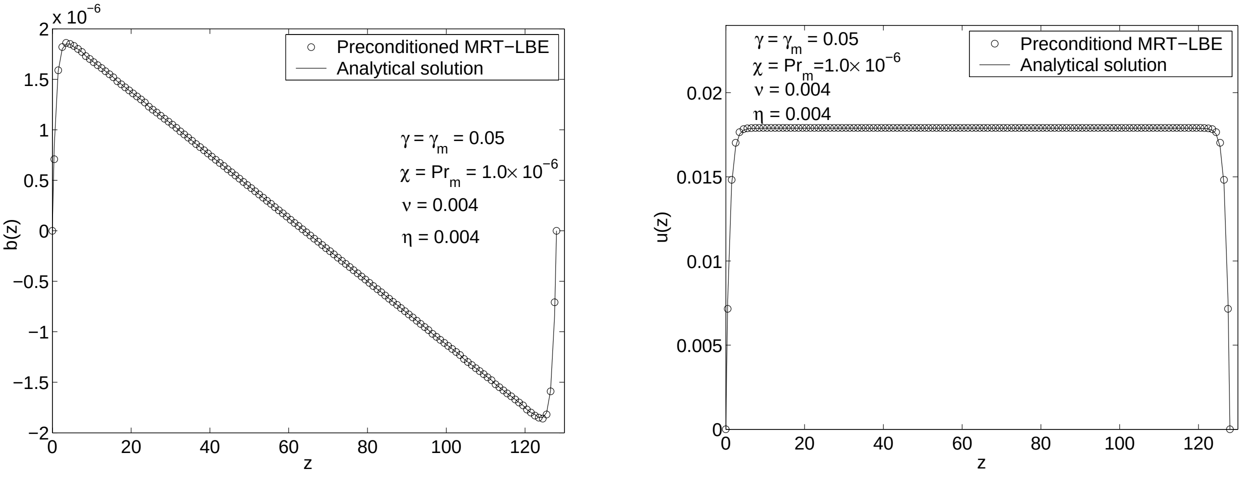


Figure 17: Magnetic field (left) and velocity (right) profiles for Hartmann flow at Ha=71.6.

Figure 18 shows the effect of the hydrodynamic preconditioning parameter γ magnetic induction preconditioning parameter γ_m on the convergence rate for two problems, Hartmann flow and a lid driven cavity with a magnetic field applied; both cases were started from rest. The residual error is a measure of the difference in the velocities at successive time steps. For cases with low γ, this is higher at first because of the accelerating effect of the preconditioning, but later on it becomes much lower as the cases with low γ approach the converged solution earlier. Both cases used $\gamma = \gamma_m$. Tests had found that setting the convergence parameters in the flow and induction equations to be the same gave the fastest convergence (when using equal viscosity and resistivity), and it can be seen that an order of magnitude decrease in compuational effort results from the preconditioning algorithm.

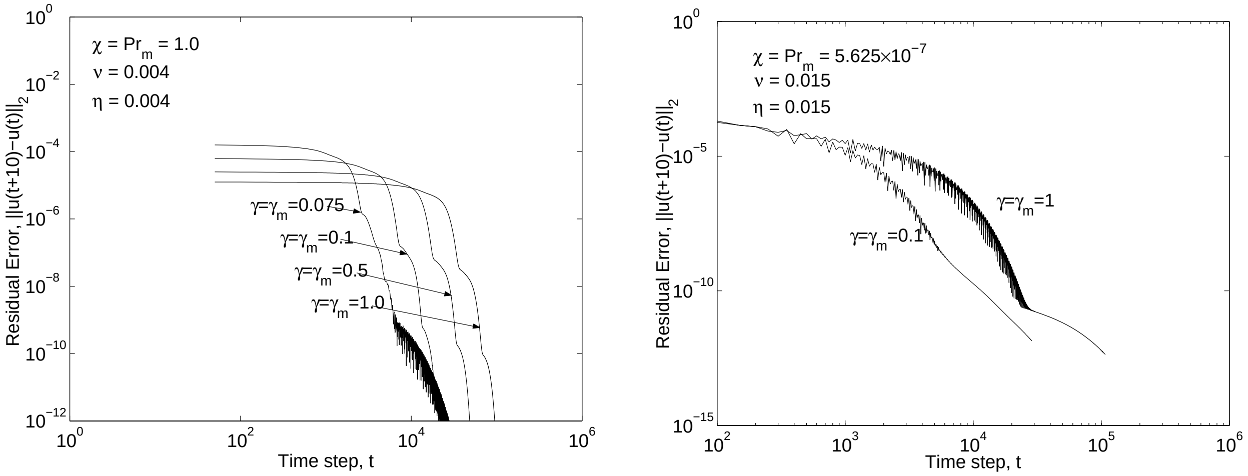


Figure 18: Convergence to steady state for Hartmann flow (left) and a 3-D lid driven cavity (right).

4.4 Validation examples

4.4.1 MHD flow in a circular pipe – Gold problem

One case for which an analytical solution is available is the laminar flow through a pipe with a magnetic field imposed in the direction perpendicular to the axis. This is often known as the Gold problem after an early paper (Gold, 1962). The arrangement is shown in Fig. 19.

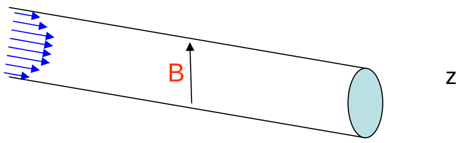


Figure 19: MHD flow in a pipe (Gold problem).

Figure 20 shows the velocity profiles for the directions parallel and perpendicular to the imposed magnetic field for a Hartmann number of 10 (based on pipe radius); 80 grid points were used to resolve one diameter. Good agreement with the analytical solution can be seen.

4.4.2 3D MHD flow in a lid-driven cubic cavity

A true 3-D MHD validation case is provided by the lid-driven cavity. In the cases presented here, the flow is in a cubic box with a the top lid moving in the x direction and the magnetic field applied either in the y or the z direction as in Fig. 21.

Figure 22 compares the results for Case 1 (field perpendicular to lid) for low and high Pr. These show velocities through the centre of the cube faces. Significant differences can be seen, demonstrating the effect of Pr on the flow. This is an important result, as most codes use a reduced formulation in

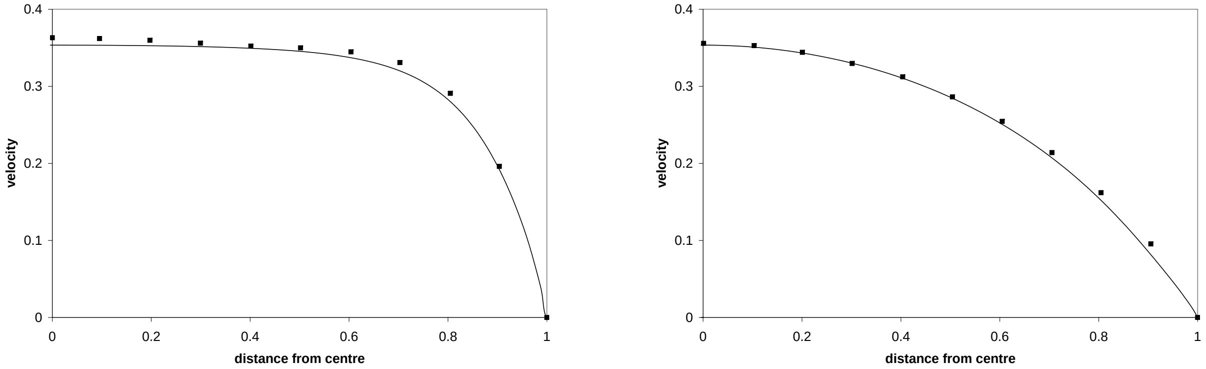


Figure 20: Velocity profiles for directions parallel (left) and perpendicular (right) to applied field (Ha=10).

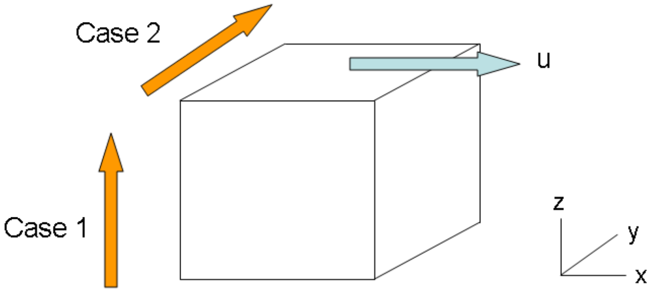


Figure 21: Schematic view of lid-driven cavity problem.

which non-linear terms in the MHD equations are neglected, corresponding to low Pr. Tests with our code showed that when the induced magnetic became comparable with the applied field, the value of Pr started to influence the results. Thus, below the value at which the induced field becomes small, the results will become insensitive to Pr. Figure 23 shows the value of u in the central plane ($z = 0.5$), which indicates the presence of side wall jets for this case.

Figure 24 compares the results for Case 2 (field in spanwise direction) with those obtained with a finite difference code (Morley et al., 2004). Good agreement can be seen, although there is a slight departure for the case with the profile of u against y; this is most likely a consequence of the resolution used in the LBM (128^2) and examination of runs with different resolutions upheld this.

4.4.3 Turbulent MHD pipe flow

This case involved the simulation of a turbulent flow through a cylindrical pipe as in Fig. 25. A fully developed flow entered a pipe; at the inlet, there was no magnetic field, but further downstream it entered a region where a field was imposed perpendicular to the pipe axis. In the simulations, a linear variation from zero to the desired strength was made over a distance of two pipe diameters. This arrangement is similar to the recent experiments at UCLA (Morley, 2006, private communication). Statistics were collected downstream to see how the field had affected the turbulence.

The inlet conditions were obtained by running a pressure-driven pipe flow simulation with period boundary conditions in the axial direction. After a statistically steady state had been reached, the flow field on one cross sectional plane was output every time step for 5000 time steps. The Reynolds number based on the bulk velocity was 4900.

The domain size used for the computation was $2400 \times 120 \times 120$ and this was run on 32 processors

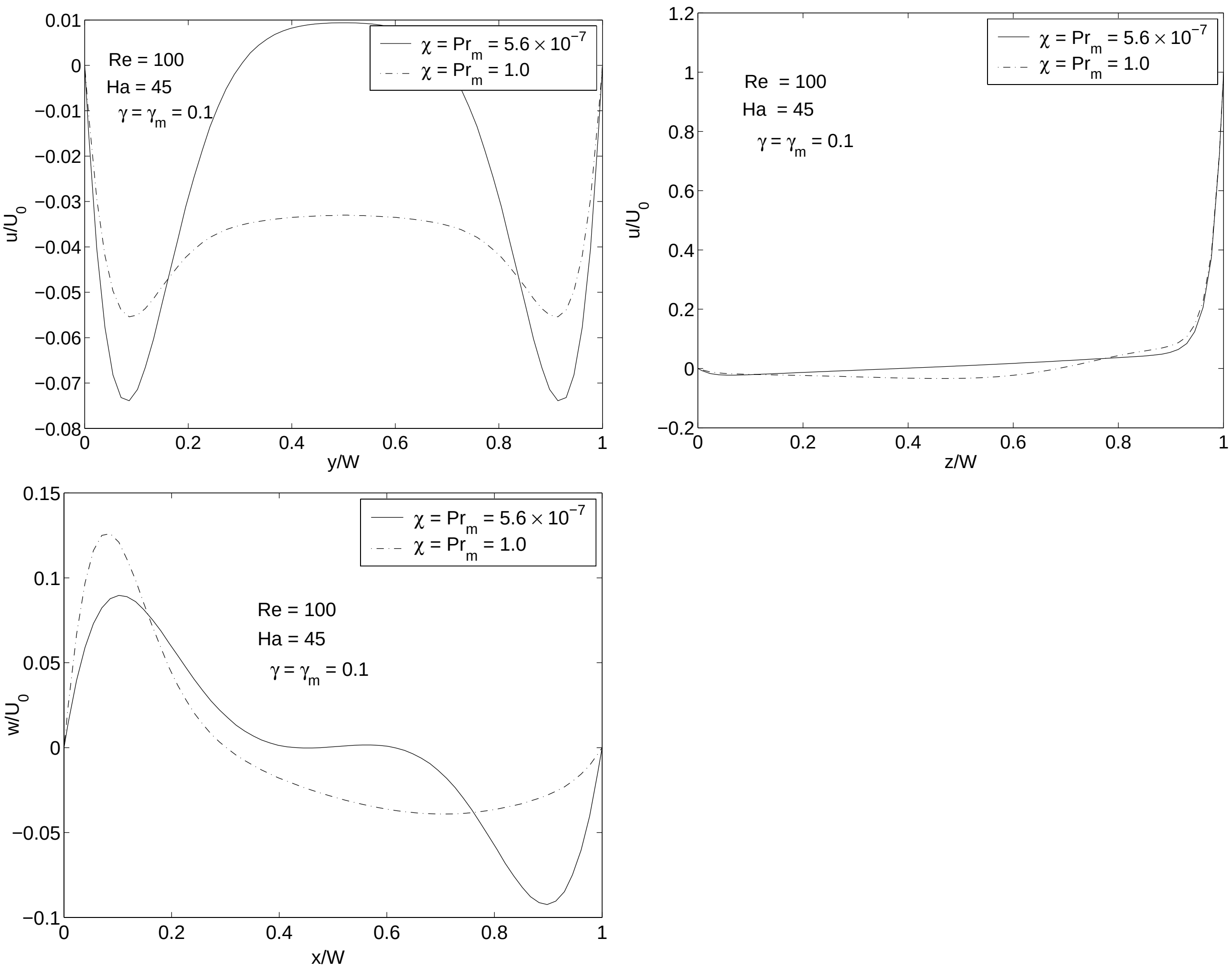


Figure 22: Comparison of results of lid-driven cavity problem (Case 1) for low and high Pr.

on the NERSC supercomputer Bassi. With this domain size, a length of twenty diameters was simulated, and Fig. 26 shows the mean velocity profile and turbulence statistics 15 diameters into the region with the magnetic field. A Hartmann number of 20 (based on the pipe radius) was used and the profiles are in the direction parallel to the field and through the pipe centre.

It can readily be seen that for the case with no MHD, the LBM overpredicts the turbulence and there is a corresponding underprediction in the velocity. Studies showed this was a resolution effect, with the error changing roughly as the square of the grid spacing. The choice of boundary conditions on the curved wall may have played a part – very little work has been done on turbulent flows in curved geometries with the LBM. While the boundary conditions are known to work well for laminar flows, the situation for turbulent flows is unclear, and there may be some staircasing effects.

The velocity profile shows some flattening, the profile is not as flat as is the case in (laminar) Hartmann flow, but it could be that with a longer pipe the profile would become flatter. At 15 diameters downstream, the flow would not be expected to be fully developed. The overall velocity appears to be slightly lower, but one should keep in mind that this is for a line through the pipe so this does not imply mass loss. All three components of turbulent fluctuations are substantially reduced and the peaks shifted away from the pipe wall, and these trends are consistent with preliminary results from UCLA (Morley, 2006, private communication).

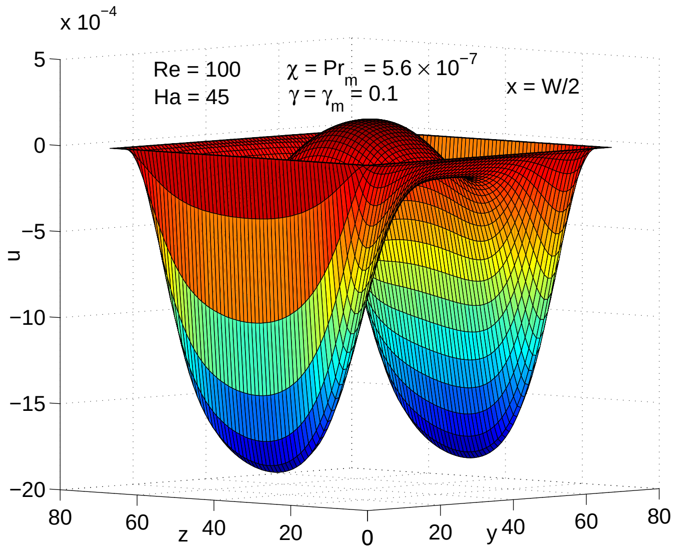


Figure 23: Plane at z=0.5 showing wall jets for Case 1 with Pr$_m$=5.6×10^{-7}; Ha=45, Re=100.

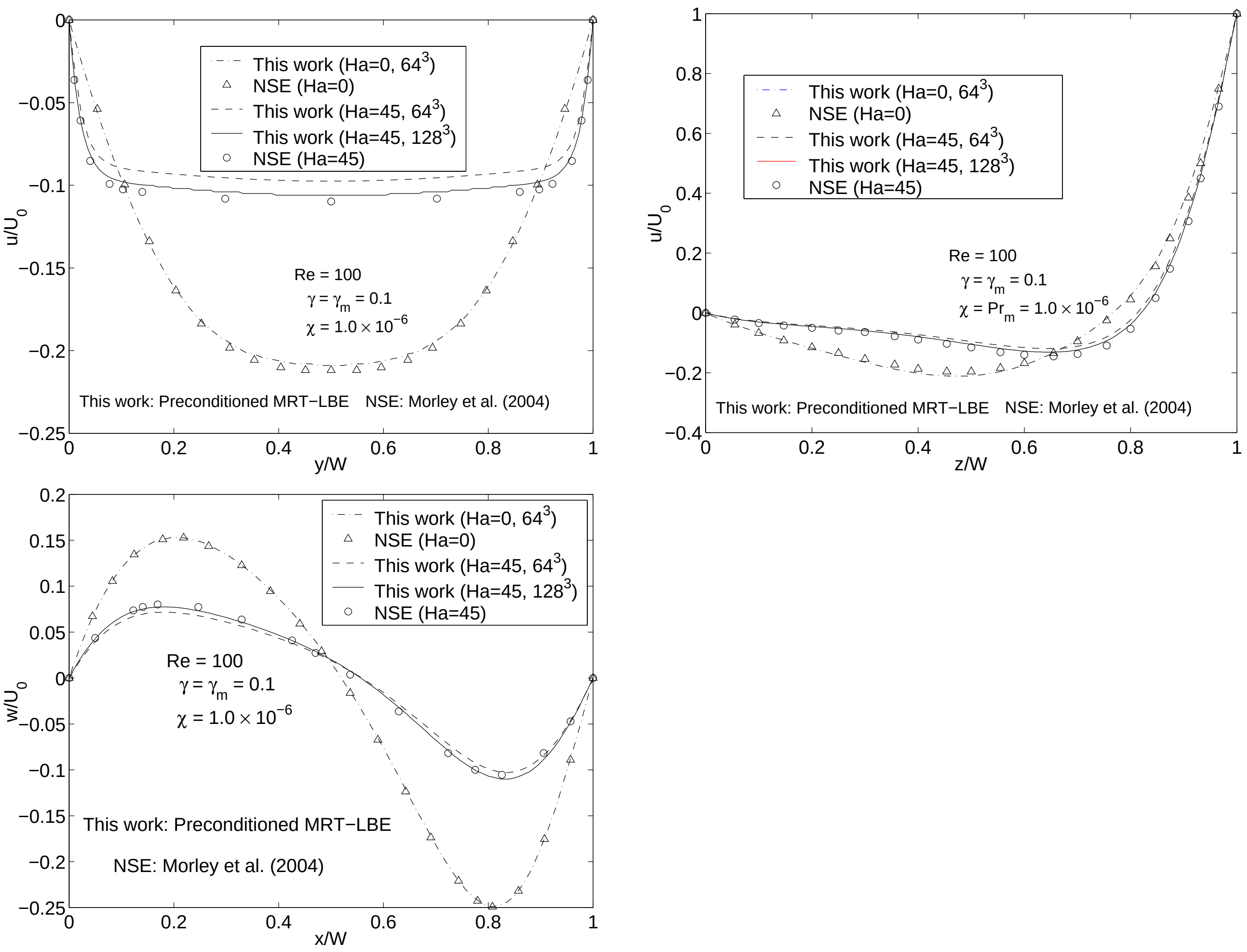


Figure 24: Comparison of results of lid-driven cavity problem (Case 2) with those of a finite difference code.

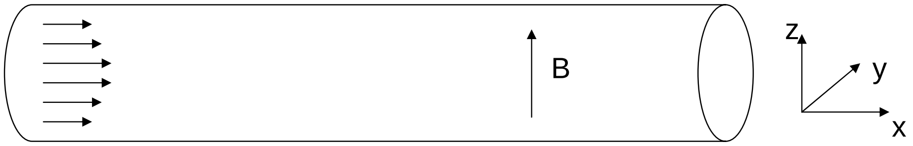


Figure 25: Schematic figure of turbulent MHD flow through a pipe.

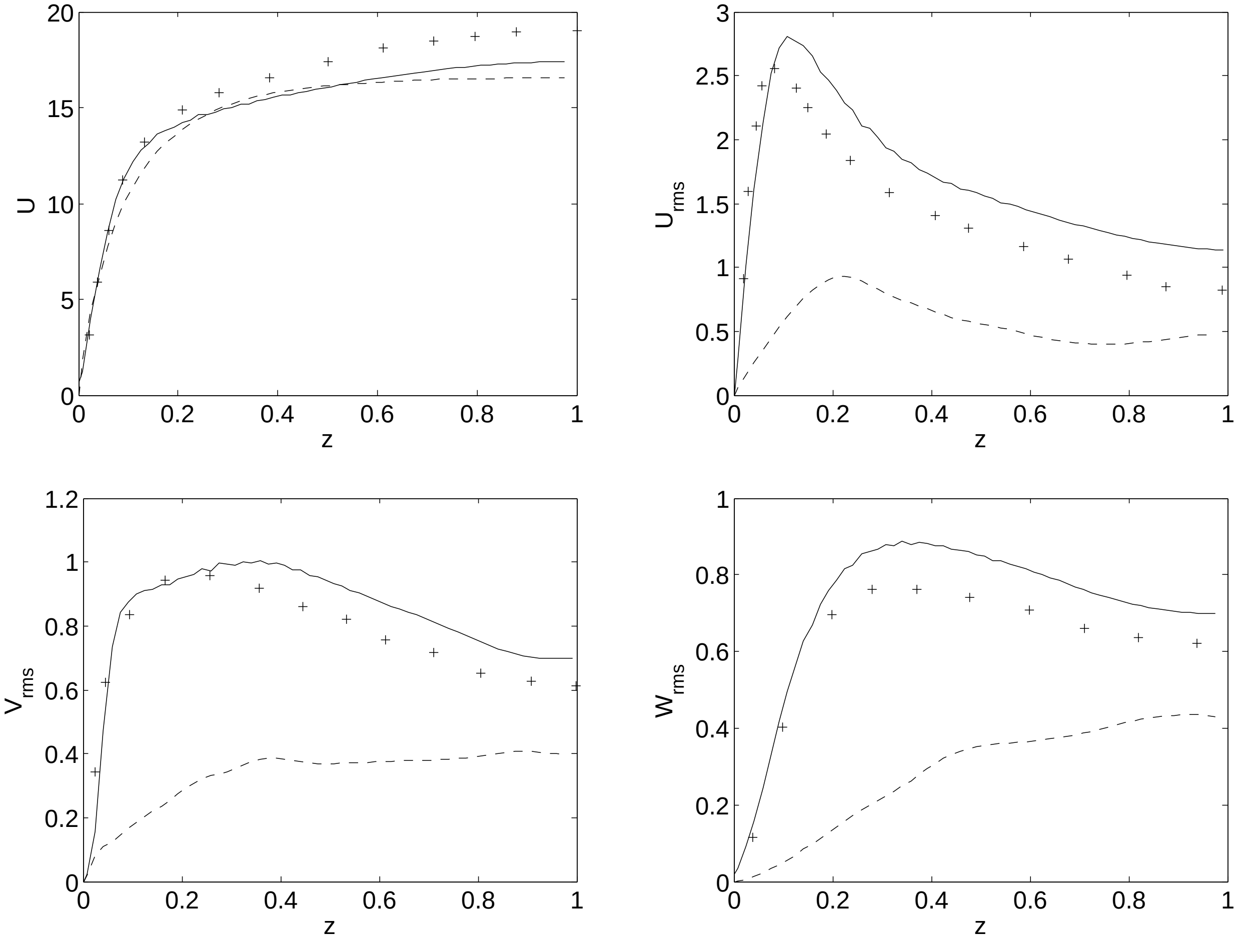


Figure 26: Profiles of mean velocity and rms velocity fluctuations along z axis through centre, taken 15 diameters from entry into the magnetic field. Solid line is LBM for no magnetic field (Ha=0), crosses DNS of Eggels et al. (1994) at Re=5300, Ha=0 and dashed line LBM with Ha=20.

5 Heat Transfer and Passive Scalar Transport

Many fluid flow problems encountered in fusion engineering, such as flows in thermal blankets or the liquid walls of fusion chambers involve the transport of large amounts of heat. In such cases, one approach to modelling the thermal transport is to solve the advection–diffusion equation:

$$\frac{\partial h}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} h) = \boldsymbol{\nabla} \cdot (\kappa \boldsymbol{\nabla} T) + \mathcal{S} \tag{47}$$

where h is the specific enthalpy, u the velocity, κ is the thermal conductivity and $\mathcal{S}$ is a source term, which may arise from, for example, chemical reactions or neutron absorption. A similar equation can be written for the transport of a fluid constituent. For the calculation of the density of a component s, ρ_s, the equation used is:

$$\frac{\partial \rho_s}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \rho_s) = \boldsymbol{\nabla} \cdot (D_s \rho \boldsymbol{\nabla} (\rho_s/\rho) + \mathcal{S} \tag{48}$$

where ρ is the total density, D is the diffusion coefficient, and $\mathcal{S}$ is again a source term. This can be due to chemical reactions, or, particularly for the case of tritium breeding, nuclear reactions.

Since Eqs. (47) and (48) are of the same general form, the same method will be used for their solution. The subsequent discussions in this section will be applicable to both the heat transfer and the mass transport equations. For simplicity, the discussion of the solution procedure will use the generic form of the advection–diffusion equation:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \phi) = \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} \lambda) + \mathcal{S} \tag{49}$$

The approach taken here was to use a hybrid approach in which the lattice Boltzmann equation was used to calculate the velocity field and a finite difference scheme for Eq. (49). Although it is possible to solve this equation directly within the LB framework, in practice problems such as numerical instabilities make a hybrid scheme the preferred choice (Lallemand and Luo, 2003b).

One of the principal motivations for using an LB solver in MetaFlow was to allow the code to run efficiently on parallel supercomputers with large numbers of processors. In order that Eq. (49) could be solved without requiring additional data packets to be sent, an explicit method was required. Although at first sight Eq. (49) appears to be very straightforward to solve, given that the velocity field is specified, it does present potential difficulties. Simple central difference schemes are unstable for low diffusivities and care must be taken to avoid the introduction of numerical diffusion in upwind differencing schemes (Press et al., 1997), which are typically employed to control numerical dispersion errors.

5.1 Solution of advection equation by TVD scheme

We employ a total-variation diminishing (TVD) scheme to represent the advection step in the transport equation, so as to maintain sharpness of profiles with controllable numerical dispersion error. The approach taken here was to start by considering a simple one-dimensional case with no diffusion or source terms. Figure 27 shows a control volume around node 0. The upstream node is denoted by the suffix W and the downstream node by E. Lower case letters are used to identify the upstream and downstream faces of the control volume. The approach taken here was to start by writing expressions for the values of ϕ on the faces, averaged over the period from t to δt:

$$\phi_w = \phi_W + (\phi_0 - \phi_W)(0.5 + \hat{u}/2) \tag{50}$$

$$\phi_e = \phi_0 + (\phi_E - \phi_0)(0.5 + \hat{u}/2) \tag{51}$$

where ϕ is calculated at t and $\hat{u} = u\delta t/\delta x$ is a dimensionless velocity, which is obtained by solving the MRT-LBM as discussed in previous sections. Note that the above uses upwind differencing and assumes flow from W to E; the formulations can easily be modified for flow in the opposite direction by switching W and E (and w and e) and reversing the sign of $\hat{u}$. There are other expressions which can be used for ϕ_w and ϕ_e, and a discussion of these, along with the principles on which the derivation of our scheme was based, can be found in (lecture notes).

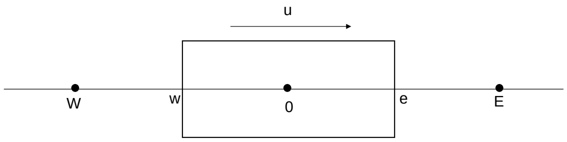


Figure 27: Control volume for scalar equation.

Considering the fluxes across the faces w and e, the value at $t + \delta t$, ϕ_0' is:

$$\phi_0' = \phi_0 + \hat{u}(\phi_w - \phi_e) \tag{52}$$

substituting the relations for ϕ_w - ϕ_e leads to:

$$\phi_0' = \phi_0 - \hat{u}(\phi_0 - \phi_W) - 0.5\hat{u}(1 - \hat{u})\left[(\phi_E - \phi_0) - (\phi_0 - \phi_W)\right] \tag{53}$$

The application of this equation to problems in can lead to numerical problems, with spurious oscillations developing in regions in which there exist large changes in the gradients of ϕ. For example, for a case initially having a top-hat profile with values of $\phi = 1$ and $\phi = 0$ inside and outside the hat, use of Eq. (53) leads to negative values of ϕ at the trailing edge. In cases where there is enough diffusion, these negative regions are damped out, but with low or no diffusion (as in Eq. (53) the ripples propagate out and grow, making the whole computation unstable.

At this point, it is useful to note that a finite scheme developed by Succi et al. (1999) for solving the scalar equation in a hybrid LB model was originally incorporated into our model. This scheme was found to be unstable for cases with low diffusion where there were sharp changes and had to be abandoned. Inspection of validations presented in the original paper revealed oscillations in such regions which grew in time, and if the validations had been continued longer, we expect that they would have failed.

The usual way to deal with this problem is to introduce flux limiter functions $\Psi(r)$ to constrain the flux at the faces of the control volume (Leveque, 1990, 2002; Toro, 1999). Equation (53) then becomes:

$$\phi_0' = \phi_0 - \hat{u}(\phi_0 - \phi_W) - 0.5\hat{u}(1 - \hat{u})\left[\Psi(r_W)(\phi_E - \phi_0) - \Psi(r_P)(\phi_0 - \phi_W)\right] \tag{54}$$

where r_W and r_P are consecutive gradient functions, defined as:

$$r_W = \frac{\phi_W - \phi_{WW}}{\phi_P - \phi_W} \tag{55}$$

$$r_P = \frac{\phi_P - \phi_W}{\phi_E - \phi_0} \tag{56}$$

Many different forms of the flux limiter Ψ have been proposed, and a comparison of some of them can be found in the paper by Suratanakavikul and Marquis (1999). As well as satisfying certain conditions to ensure physically realistic behaviour, the function should be such that it preserves sharp changes in ϕ and avoids numerical diffusion without leading to instabilities. For this work, the superbee limiter was chosen:

$$\Psi(r) = \max[0, \min(2r, 1), \min(r, 2)] \tag{57}$$

Up to now, we have just considered one dimensional flow. In the implementation in our three dimensional code, ϕ was updated by applying Eq. (54) in turn in all the coordinate directions, updating ϕ after each step.

5.2 Incorporation of diffusion and source terms

Equation (54) does not contain the diffusion or source terms. However, these may be added relatively easily, through a central differencing for the diffusion term and local explicit values for the source term. The following equation is used for the remaining terms and also includes a correction term for compressible flows:

$$\begin{aligned}
\phi^{t+dt}_{i,j,k} = \phi'_{i,j,k} \quad & + \quad \Big(k_{i+\frac{1}{2}}(\lambda_{i+1,j,k} - \lambda_{i,j,k}) + k_{i-\frac{1}{2}}(\lambda_{i-1,j,k} - \lambda_{i,j,k})\Big)\, dt/dx^2 \\
& + \quad \Big(k_{j+\frac{1}{2}}(\lambda_{i,j+1,k} - \lambda_{i,j,k}) + k_{j-\frac{1}{2}}(\lambda_{i,j-1,k} - \lambda_{i,j,k})\Big)\, dt/dy^2 \\
& + \quad \Big(k_{k+\frac{1}{2}}(\lambda_{i,j,k+1} - \lambda_{i,j,k}) + k_{k-\frac{1}{2}}(\lambda_{i,j,k-1} - \lambda_{i,j,k})\Big)\, dt/dz^2 \\
& + \quad \Big[\left(\rho_{i+1,j,k}(u_{i+1,j,k} - u_{i,j,k}) - \rho_{i-1,j,k}(u_{i-1,j,k} - u_{i,j,k})\right) / (2dx) \\
& + \quad \left(\rho_{i,j+1,k}(u_{i,j+1,k} - u_{i,j,k}) - \rho_{i,j-1,k}(u_{i,j-1,k} - u_{i,j,k})\right) / (2dy) \\
& + \quad \left(\rho_{i,j,k+1}(u_{i,j,k+1} - u_{i,j,k}) - \rho_{i,j,k-1}(u_{i,j,k-1} - u_{i,j,k})\right) / (2dz) \Big] \frac{\phi_{i,j,k}}{\rho_{i,j,k}} \\
& + \quad \mathcal{S} dt
\end{aligned} \tag{58}$$

where $\phi'_{i,j,k}$ is the value after performing the advection step in the three directions. Some care needs to be taken in the calculation of the diffusion/conduction term where k changes. It was found that calculating the value of k as

$$k_{i+\frac{1}{2}} = 2\left[\frac{1}{k_{i+1,j,k}} + \frac{1}{k_{i,j,k}}\right]^{-1} \tag{59}$$

avoided any problems at discontinuities. In the implementation of Succi et al.'s (1999) model, it had been found that Succi's model was only suitable for constant k and led to severe problems when there were discontinuities, but when the diffusion term was cast in the way described here, it worked well. However, as mentioned previously, there were other problems with Succi et al.'s method.

5.3 Validation examples

This section presents some of the cases used for testing the code. The validations presented cover all aspects of scalar transport, advection, diffusion and reaction, and include a few standard problems where data from comparison are available.

Diffusion Test Case

A test to check on the diffusion part of the formulation was undertaken in which the evolution of a step profile over time was calculated. This problem considered the diffusion of a tracer whose concentration was initially unity for $x < 0$ and zero for $x > 0$. The diffusion coefficient was 0.1 when normalised with the time step and grid spacing, and the simulation was run for 100 000 iterations. Figure 28 compares the computed profiles with those from the analytical solution, which is (Arcidiacono et al., 2006):

$$\rho_s = \frac{1}{2} + \frac{\Delta \rho_s}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{4Dt}}\right) \tag{60}$$

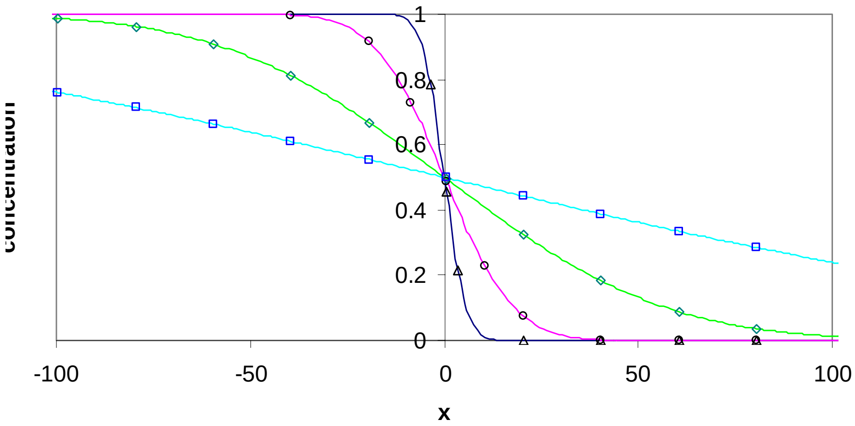


Figure 28: Evolution of temperature profile over time. Results for 100 (triangles), 1000, 10 000, and 100 000 time steps (squares). Lines are MetaFlow predictions and symbols are values from analytical solution.

The excellent agreement between the code results and the analytical solution verifies the suitability of the solution scheme.

Advection with no Diffusion

An important test of any advection/diffusion solver is to see whether it can preserve sharp changes. A useful test is to start with a top hat profile in a uniform velocity field and track the evolution of the concentration or temperature for the case of zero diffusion/conduction. Some smoothening at the sides is inevitable, but it is important to check that the central region does not decay. One example is illustrated in Fig. 29. In this case a square region of side 13 grid points centred on (106.5, 56.5) was assigned an initial temperature of 400K, 100K above that of the rest of the domain. The flow field had a velocity of 0.2 (normalised with the grid spacing and time step) in both the x and y directions and the simulation was run for 500 time steps during which the hot region moved 100 grid points in each direction. The central region can be seen to be still flat, with a peak temperature of 399.8K, which is what is desired.

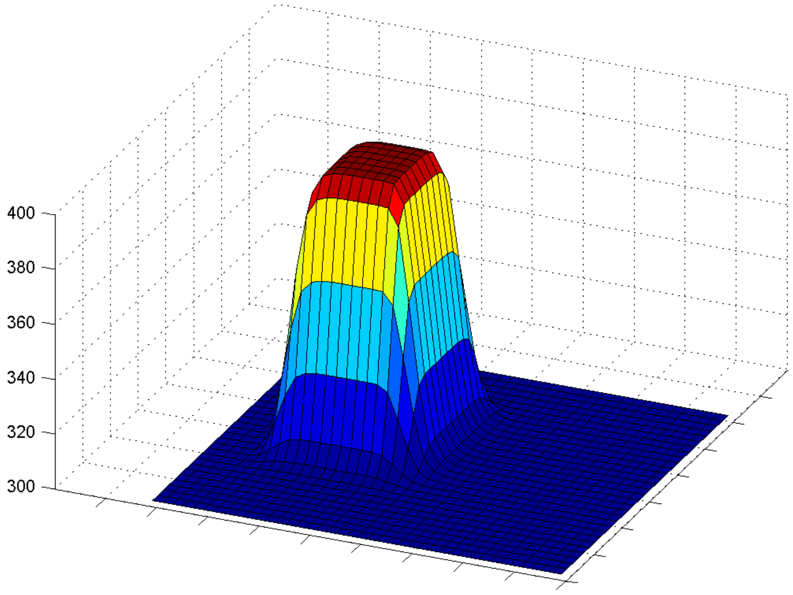


Figure 29: Temperature field after 500 time steps.

Flow Between two Heated Plates

A classic problem in heat transfer is that of a fully developed flow entering a region where it is subjected to heating. The case undertaken in this study is illustrated in Fig. 30; in this case, the flow was between two parallel walls and a constant heat flux was applied by the heating elements. The flow entered the heated region with a parabolic velocity profile and constant temperature, and temperature profiles 5, 10 and 15 channel widths downstream from the start of the heated zone are shown in Fig. 31. The channel width was resolved with 64 grid spacings. Results from MetaFlow are compared with an analytical solution for large z (Bird et al., 2002), and it can be seen that the agreement far downstream is excellent.

Taylor Dispersion

If a miscible fluid is suddenly injected into a fluid flowing through a channel, a small region with a high concentration of the solute will result. As this region travels along the channel, it gradually broadens. The axial dispersion is a result of both diffusion and (for non-uniform velocity profiles) convection, since the diffusion causes the solute to move to regions with different velocities. This case is known as Taylor dispersion and is discussed by Deen (1998). It can be shown that, for sufficiently long observation times, the dispersion coefficient is given by:

$$K = D\left(1 + \frac{\mathrm{P}e^2}{C_0}\right) \tag{61}$$

where C_0 is a constant which depends on the flow profile and the Peclet number, Pe=U_0H/D is defined in terms of the peak velocity U_0, the channel width, H, and the molecular diffusion coefficient D. The term involving Pe reflects the effect of hydrodynamic stretch resulting from the flow shear, and at high

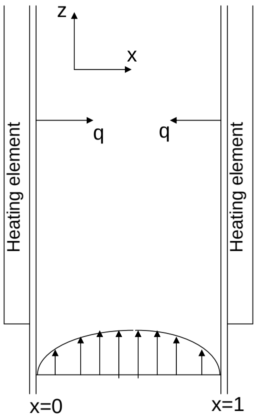


Figure 30: Flow between parallel plates. Fluid enters with parabolic velocity profile and constant temperature.

Pe is the dominant term.

A series of numerical experiments were performed in which a point source was introduced into the centre of a square duct in which there was fully developed flow. Figure 32 shows the evolution of the concentration field for a case with Pe=92. As the introduced fluid diffuses away from the centre, it moves into regions with lower velocity leading to the comet tail appearance.

For a quantitative test of the dispersion, the code was set to calculate variance of the concentration in the axial direction $\sigma^2 = \int (x - \overline{x})^2 c$ where c is the proportion of introduced fluid at x. This can be related to the dispersion coefficient through $K = \sigma^2/2t$. According to Eq. (61), the relative dispersion, $K/D - 1$, should vary linearly with Pe^2 and this is indeed seen in Fig. 33. The constant C_0 is the ratio of the value of the ordinate to the abscissa and this was found to be about 530, the numerical computations gave values ranging from 517 to 551, a very small variation when one considers that the simulations spanned an order of magnitude change in Pe.

Falling Liquid Film with Chemical Reaction

Another validation considered the case of absorption of a gas into a falling liquid film and its subsequent reaction. This is illustrated in Fig. 34 where the liquid film enters a region where there is a gas present which dissolves in it and then undergoes an irreversible first order reaction. A example of such a situation would be the absorbtion of carbon dioxide by a concentrated aqueous solution of sodium hydroxide. Figure 35 shows the concentration profiles of the absorbed gas at different positions downstream from the entrance point. The computed profiles are compared with the analytical solution given by Bird et al (2002) for large z

$$\frac{c}{c_0} = \frac{\cosh\left[\sqrt{k_1 L^2/D}(1 - x/L)\right]}{\cosh\left[\sqrt{k_1 L^2/D}\right]} \tag{62}$$

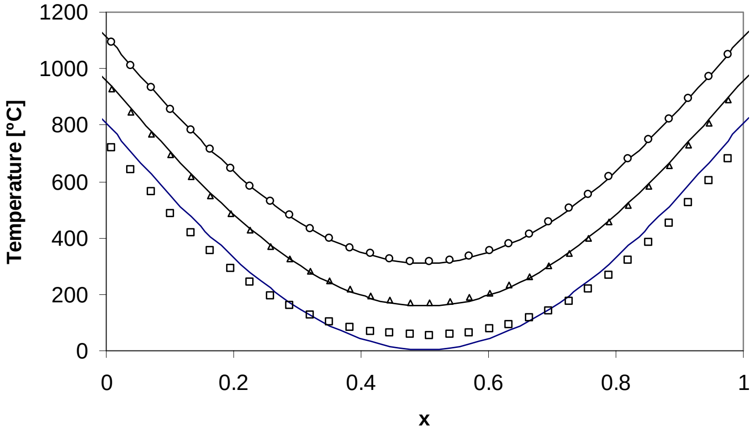


Figure 31: Temperature profile across channel. Results for 5, 10 and 15 channel widths. Lines are MetaFlow predictions and symbols are values from analytical solution for large z.

where k_1 is the rate constant for the reaction with the liquid. The agreement at large z can be seen to be very good.

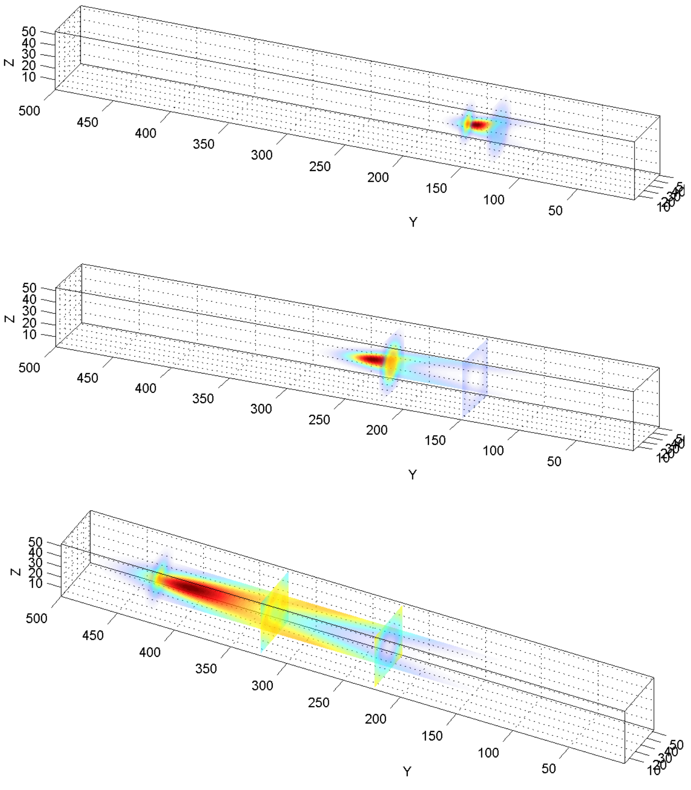


Figure 32: Concentration profiles after 2200, 4400 and 8800 time steps for Pe=92.

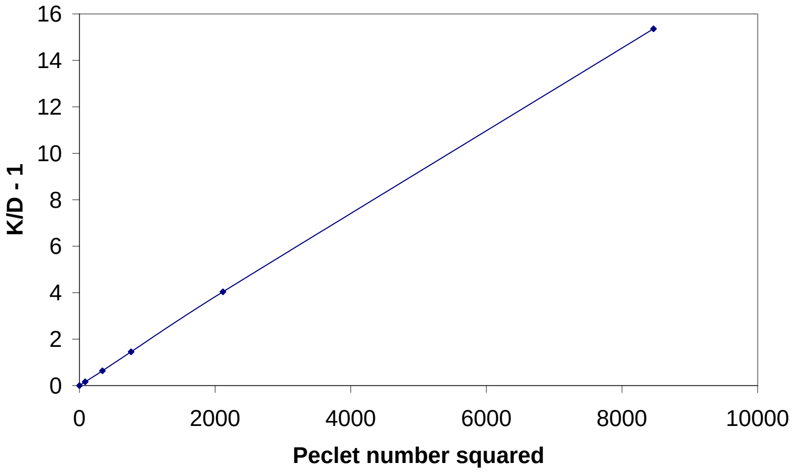


Figure 33: Concentration profiles after 2200, 4400 and 8800 time steps for Pe=92 Variation of relative dispersion with Péclet number.

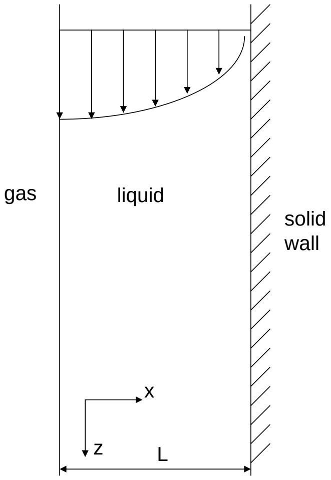


Figure 34: Falling liquid film with absorbtion of species from gas.

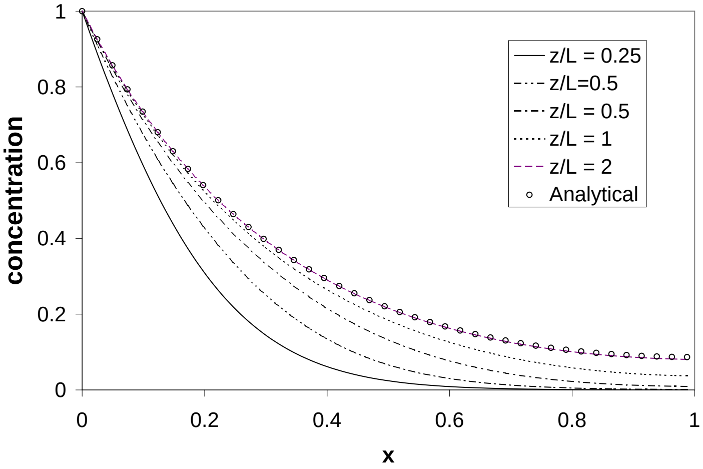


Figure 35: Concentration profiles at different stations downstream. Concentrations normalised to unity at surface.

6 Non-Uniform Grid Stretching in LBM to Resolve Thin Hartmann Layers

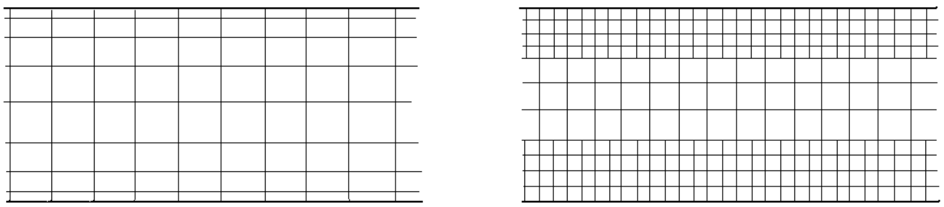
The use of nonuniform grids is desirable in many applications and is important in regions of flow where sharp gradients are present. Magnetohydrodynamic flows at high Hartmann numbers provide examples where stretched grids are likely to result in very high savings in computational time. The regions with high gradients are contained within very narrow layers close to the wall, with the flow being relatively constant outside these layers. Figure 36a shows an example of a stretched grid where higher resolution is achieved in the wall-normal direction near the wall.

In their basic form, the LBE formulations are restricted to uniform grids in that the minimum streaming distance of the particle populations in one time step is exactly equal to the minimum lattice spacing. In other words, the discretization of the configuration and particle velocity spaces are coupled. This lockstep advection of particle populations is a feature inherited from its predecessor, i.e. the lattice-gas automaton (LGA) (Wolf-Gladrow, 2000) and is not necessary for the LBE. It was realized that the LBE are actually simplified forms of the Boltzmann equation and hence can be solved without the coupling of the physical and particle velocity spaces and can in principle be implemented on any mesh (Cao et al., 1996). Thus, it was proposed by He et al. (1996) that the collision step can be computed locally on the lattice grids in the usual manner; after the collision step, the particle distributions move according to their velocities $\vec{e}_\alpha$; although the advected distance of the particle distributions may not, in general, coincide with the mesh spacing (as shown in the example in Fig. 37). The distribution functions in these locations can always be computed using interpolation; after interpolation, the collision and streaming steps are repeated. It has been shown that, if the interpolation method is at least of second-order, the Navier-Stokes equations can still be recovered from LBE (He, 1997). Thus, the order of interpolation schemes should not be lower that the order of accuracy of the LBE.

To improve the computational efficiency, here we employ a variant of this original interpolation-supplemented LBE, in which the streaming and the interpolation steps are carried out in two distinct steps. Here, we combine them in one step - the *interpolated-streaming step*. Also, the original approach was applied to the SRT-LBE. However, inspection of the method shows no reason why it cannot be applied in a similar manner to the MRT and the induction equation, and this is indeed what was done in this project. A second-order Lagrange interpolation was employed in the implementation, which is are carried out in the respective upwind directions for the particle velocity directions. An example of this implementation for one particle velocity direction, including the interpolation coefficients in 3D, is given in Appendix I.

An alternative method for variable grid spacing is what we shall term the *multiblock* method, in which local grid refinement is applied in regions requiring higher resolution, such as around boundaries. This is illustrated in Fig. 36b, where the grid spacing is reduced by a factor of two in the regions close to the walls. There is also a change in the timestep used so that the coarse region uses a step as twice as long as the fine region. A second order interpolation scheme is used to ensure continuity between the coarse and fine regions, and this is discussed further by Fillipova and Hänel (1998) and Lu et al. (2002). Transformation relations for the distribution functions on the fine and coarse grids. Although this approach allows higher resolution in regions of interest, it does not permit non-cubic grids.

In constrast, the ISLBM does allow non cubic grids, though it is not suitable for stretching grids in any coordinate direction relative to the others, i.e. for non-rectangular meshes. However, this does not pose a problem for many cases of practical interest. On the other hand the need to use the same, small grid spacing in the streamwise and spanwise directions, would limit the effectiveness of local grid refinement for 3-D applications. Moreover, the ISLBM is relatively easy to implement and naturally amenable for parallel implementation and for this reason was employed in this project.



(a) Variable grid for use with ISLBM (b) Locally refined grid

Figure 36: Different types of grids for the LB method.

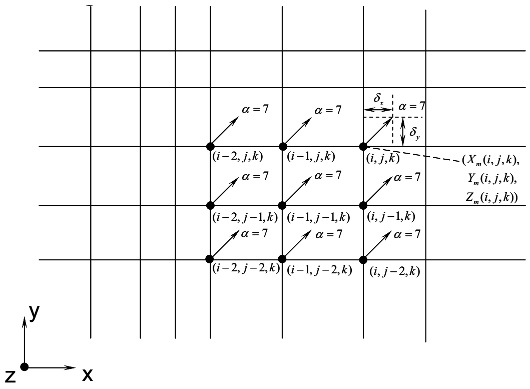


Figure 37: Illustration of a second-order upwind Lagrangian interpolation for a particle velocity direction (α= 7) to implement non-uniform grids in the LBM.

6.1 MHD Flow between parallel plates at high Hartmann number

Following preliminary tests with non-MHD flows to ensure the ISLBM was working correctly, a series of studies was carried out with MHD flows. The case described here was a simple Hartmann flow between two insulating parallel plates, where the magnetic field is applied perpendicular to the plates. For this case, there is an analytical solution for the velocity v and the induced magnetic field B_i

$$v = \frac{dP}{dx}\frac{L}{B_0}\sqrt{\frac{\rho}{\sigma\nu}}\left[\frac{1-\cosh(\mathrm{Ha}\,z/L)/\cosh(\mathrm{Ha})}{\tanh(\mathrm{Ha})}\right] \tag{63}$$

$$B_i = \frac{dP}{dx}\,\frac{\mu_0\rho L}{B_0}\,\left[\sinh(\mathrm{Ha}\,z/L)/\sinh(\mathrm{Ha}) - z/L\right] \tag{64}$$

where dP/dx is the driving pressure gradient, ρ density, σ electrical conductivity, ν kinematic viscostity and μ_0 magnetic permeability. The coordinate z is defined such that $z = -L$ on the lower plate and $z = L$ on the upper plate. The Hartmann number is defined as:

$$\mathrm{Ha} = B_0 L\sqrt{\frac{\sigma}{\rho\nu}}$$

The solution is discussed in depth Müller and Bühler (2002), who also give the more general solution for finite conductivity walls and a square duct. Note that at very high Hartmann numbers, computer code may fail to solve the above equation, since the terms inside the hyperbolic functions become to large. In this situation, approximate (asymptotic) forms for high Ha can be used (this is also addressed by Müller and Bühler).

The grid used should be set such that several points lie within the Hartmann layer, with the grid point separation increasing to large values outside this region. One suitable formulation is due to Roberts (1971) in which a uniform grid (i.e. with equally spaced points) $\overline{z_k}$ in the range $0 < \overline{z_k} < 1$ is transformed into a stretched grid z:

$$z = 2L\frac{(\beta + 2\alpha)\left[(\beta+1)/(\beta-1)\right]^{(\overline{z}-\alpha)/(1-\alpha)} - \beta + 2\alpha}{(2\alpha+1)\left\{1 + \left[(\beta+1)/(\beta-1)\right]^{(\overline{z}-\alpha)/(1-\alpha)}\right\}} \tag{65}$$

If the parameter α is used to control where the points cluster. With $\alpha = 0$, only points near $z = 2L$ are spaced closely; with $\alpha = 0.5$, points are closely spaced at regions near $z = 0$ and $z = 2L$. The parameter β determines the degree of stretching, and Smolentsev et al. (2005) suggest using a value

$$\beta = \left[\frac{\mathrm{Ha}}{\mathrm{Ha}-1}\right]^{0.5}$$

In practice, this was found to give grids in which the spacing was far too small near the walls. The above recommendation was based on the assumption that that the dimensionless boundary layer thickness (corresponding to the Hartmann layer in this case) is ∼1/Ha. Inspection of profiles generated suggested that a the boundary layer thickness, δ/L, was actually ∼5/Ha, though the definition of the end of the layer is somewhat arbitrary.

In the Hartmann flow computations presented here, the relation used was modified to become:

$$\beta = \left[\frac{\mathrm{Ha}/5}{\mathrm{Ha}/5-1}\right]^{0.5}$$

With this factor, flow velocities predictions within 1% of the analytical solutions could be obtained by the lattice Boltzmann code.

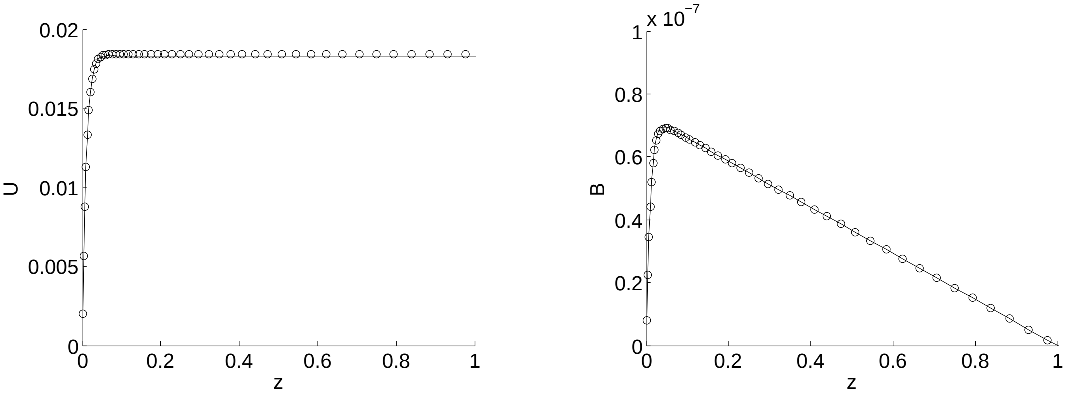


Figure 38: Velocity (U) and magnetic field (B) profiles for Ha=100; line LBM, circles analytical. 96 grid points.

Figures 38–40 show the velocity and induced magnetic field profiles for Hartmann numbers spanning two orders of magnitude of 100, 1000, and 10 000. The coordinate z in these figures runs from 0 at the wall to 1 at the centre. Figure 38 uses a linear scale, but the other two use a logarithmic scale for the abscissa since the Hartmann layers are very narrow in these cases. The comparison with the analytical solution is very good, and all cases predicted the centreline velocity to within 0.5%; the locations of the circles correspond to the grid points so it can be seen that about 10 points are used to resolve this layer.

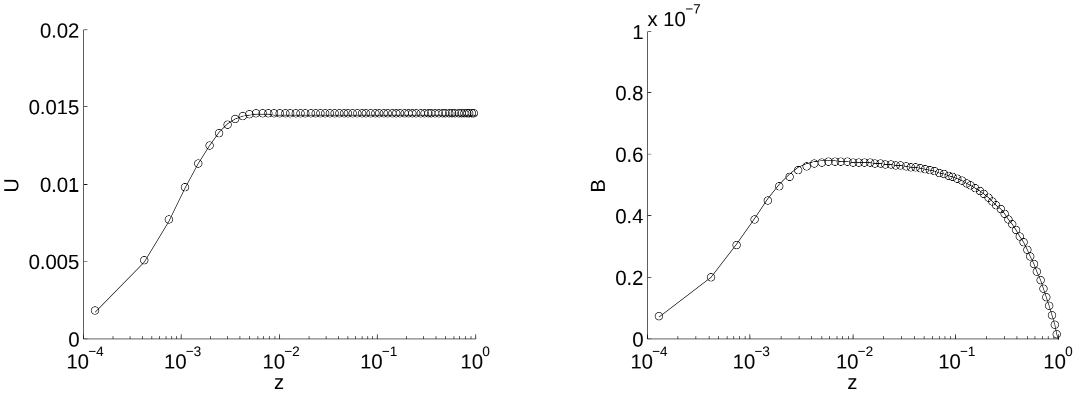


Figure 39: Velocity and magnetic field for Ha=1000; line LBM, circles analytical. 128 grid points. Log scale used for abscissae.

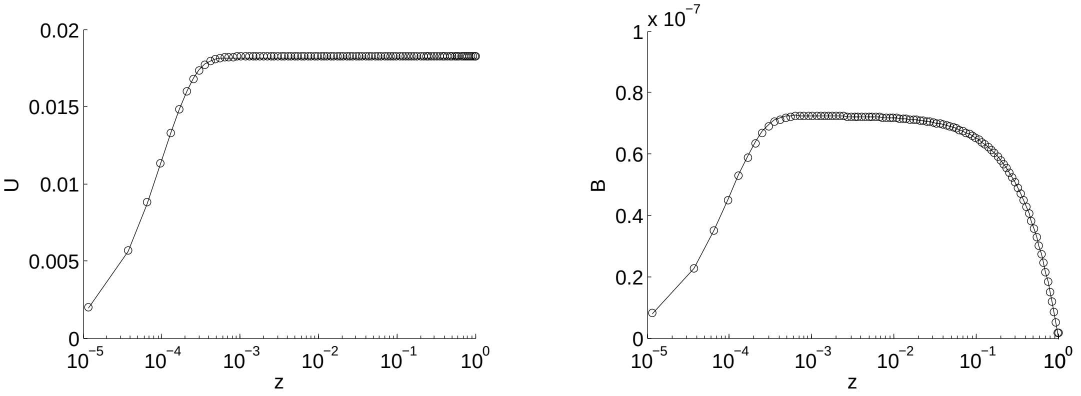


Figure 40: Velocity and magnetic field for Ha=10 000; line LBM, circles analytical. Log scale used for abscissae. 192 grid points.

In all cases, the use of a stretched grid considerably reduced the number of grid points by an order of magnitude or more. For the Ha=10 000 case, use of a uniform grid would have required 85 000 points to achieve the same near wall resolution, but in this simulation only 192 points were used to span the channel (total was actually 196 points because extra points are needed for bounding walls) – a drastic saving in computer resources.

6.2 MHD flow in insulating duct at high Hartmann number

Another test case used was the flow through a square duct with insulating walls at a high Hartmann number (in this case Ha=500). Duct flow velocity profiles are again characterised by a flat central region with narrow layers with high gradients near to the walls. The width of the layers adjacent to the walls perpendicular to the applied field direction (the *Hartmann* layers) varies as $1/\mathrm{Ha}$, but for those parallel to the field (the *side* layers), the width varies as $1/\sqrt{\mathrm{Ha}}$. The case was resolved by using 128 grid points in each direction, and Fig. (41) shows the velocity profiles obtained with MetaFlow along the wall bisectors along with the analytical solutions. The velocity at the centre differs from the analytical solution by about 2%, rather more than that in the earlier simulations. The reason for this is that the ideal grid distribution required for the y and z directions are rather different and the stretching profile used was a compromise. Use of a few more grid points in the y direction would be expected to significantly reduce the discrepancy.

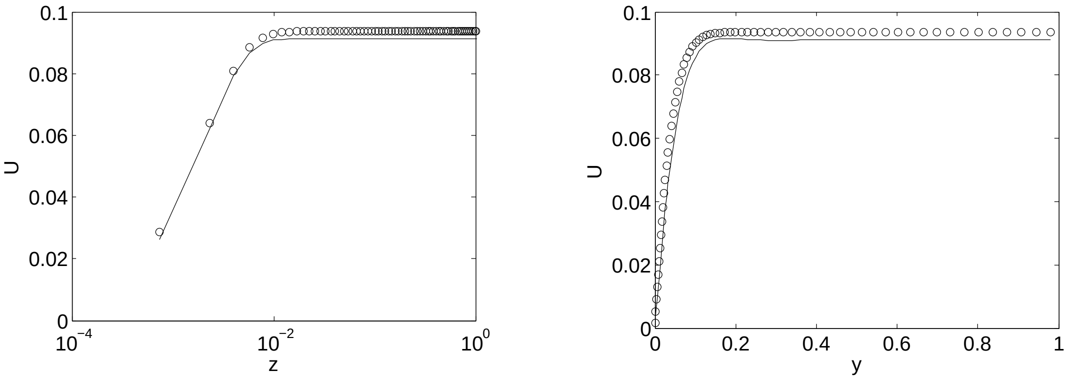


Figure 41: Velocity profiles along wall bisectors for duct case at Ha=500, left: side layers and right: Hartmann layer; line LBM, circles analytical solution. Field applied in z direction. Note that for clarity the right graph has log scale for the abscissa, and axes are normalised by the duct half-width.

The profile through the side walls was calculated with the following approximate formula, taken from Moreau (1990)

$$u = u_0(1+\eta^2)(1-\mathrm{erf}(\eta/\sqrt{2})) - \eta\sqrt{2/\pi}\exp(-\eta^2/2) \tag{66}$$

where η is a dimensionless distance $\eta = y\sqrt{\mathrm{Ha}/2}$, with y being the distance from the wall, normalised with the duct half width. The velocity u_0 is the velocity at the centre for the equivalent case with no side walls. This solution is also given a thorough discussion by Müller and Bühler (2002), but readers are warned that during the course of this analysis, the equation given in their book was found to contain a typographical error.

Two dimensional plots of field and the velocity are shown in Fig. (42). Over the central region, the velocity profile can be seen to be very flat, with the induced field having a constant gradient. These observations are in keeping with expected trends.

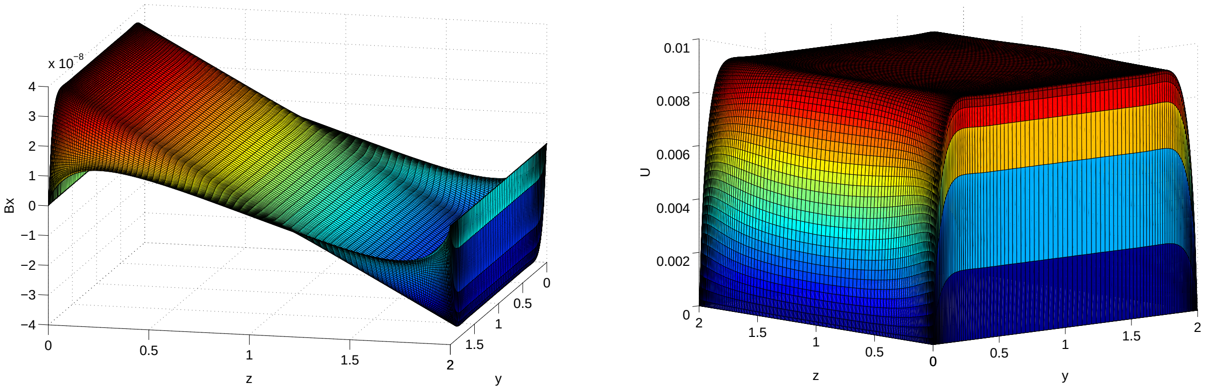


Figure 42: Induced field (Bx) and velocity (U) plots for square duct case at Ha=500.

7 Input of Complex Geometry

Two methods for specifying the geometry to be simulated are available for the MetaFlow code. The first is for the user to supply an ascii file sequentially listing all the grid points and specifying (by means of an index) which material is at each point. This is a very easy and useful method for simple geometries, as are often used for validation cases. However it does have shortcomings – information on the exact location of boundaries cannot be supplied, so the code simply uses the bounceback scheme for solid/fluid boundaries, which is equivalent to taking the wall to lie halfway between grid points and can lead to inaccuracies where walls do not align with coordinate axes. Also generation of such a file for complex geometries is not easy.

The other method, which is expected to be more commonly used, is to import the geometry from a CAD file. After consideration of the file formats available, the one selected for use by MetaFlow was the STL format, since virtually all CAD suites either produce this format directly, or the files can be subsequently converted into STL form. In an STL file, the surface of a body is rendered as a series of triangles – a simple example is shown in Fig. 43. For simple geometries only a few triangles are required, for example a cube can be constructed with just 12; for complex shapes many more triangles may be used, especially if high resolution of curved surfaces is needed.

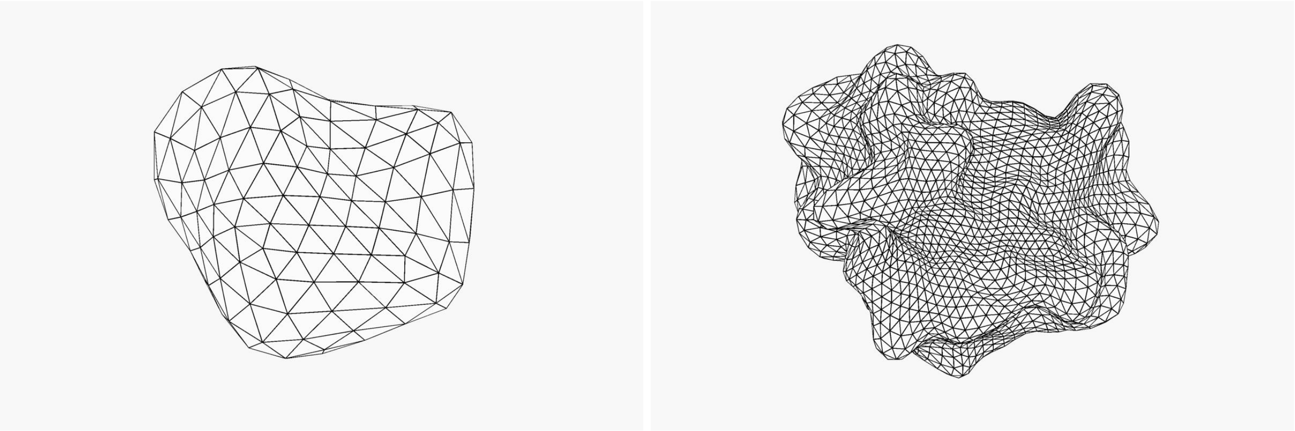


Figure 43: Illustration of surface triangulations as used in STL files.

The method used to generate the grid and boundary information is best illustrated with a simple case. Consider the situation in Fig. 44, where the flow through a pipe is to be modelled. The user would supply an STL file with the pipe geometry and specify the boundary of the computational domain (dashed line in the figure). In most cases, the domain need only extend a very short distance beyond the sides. The user must also specify a point within the pipe and a point outside and the material there (fluid or solid). In the general case, the user will specify the material in each continuous region by specifying a point in that region and the index of the material there, except that there is an option to set all unspecified regions to a particular material.

The procedure followed by the code is:

1. Code opens STL file, checks if it is in ascii or binary format and reads in triangles.

2. A Cartesian grid is generated to fit the size of computational domain requested – spacing is user defined

3. All grid points are checked to see if they are on the surface of a triangle (in practice within a set tolerance of it). These points are set aside to be dealt with later.

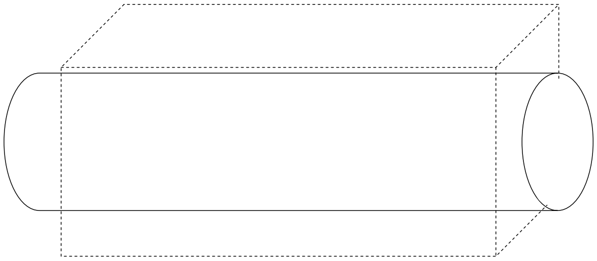


Figure 44: Schematic diagram showing cylinder and computational domain (dashed line).

4. Starting from the user-specified point, neighbouring grid points are checked to determine whether they are in the same continuous region. This is done by finding whether a line from one grid point to the other passes through a triangle. This procedure is repeated for all neighbouring points found to be in the same region, until all the points within the region are found.

5. Step 4 is repeated for all the other continuous regions.

6. Points found to be on the boundary in step 3 are assigned to neighbouring regions. Where they lie between a fluid and solid region, they are assigned to the fluid region.

7. On grid points adjacent to solid regions, the distances from the wall required for the boundary condition routines are calculated.

8 Example of 3D MHD Simulation for a Thermal Blanket Section

One case undertaken that is relevant to a current fusion engineering problem was the simulation of a section of a thermal blanket (e.g. Smolentsev et al., 2006). A cross-section of the geometry for the problem tackled is shown in Fig. 45 (Morley, 2006, Private Communication). The widths of the layers are assumed to be uniform around the perimeter, and the cell is 1.4m long. Liquid lead/lithium flows through the central region which is surrounded by the flow channel insert (FCI), a layer of a silicon carbide composite designed to provide thermal and magnetic insulation. This is in turn surrounded by a layer of Pb/Li with the outer casing being ferritic steel.

Conditions and thermophysical properties were set to be similar to those in the design. The simulation assumes the mean Pb/Li inlet velocity to be 0.1ms^{-1} with a temperature of 400°C. It it subjected to heating from the neutron flux, taken to be

$$\dot{q}''' = 3.92\exp(-0.35y) + 2.1 \qquad \text{Ferritic Steel} \qquad (67\text{–a})$$
$$\dot{q}''' = 14.33\exp(-0.90y) + 4.0 \qquad \text{PbLi} \qquad (67\text{–b})$$
$$\dot{q}''' = 5.31\exp(-0.52y) + 1.6 \qquad \text{SiC} \qquad (67\text{–c})$$

with the heating rate $\dot{q}'''$ in MW and the distance from the front wall, y in cm. The heat transfer coefficient at the outer walls was taken to be 4000Wm^{-2}K^{-1} with the external temperature rising linealy from 305–375°C for the front wall; for all other walls, a temperature of 375°C was used.

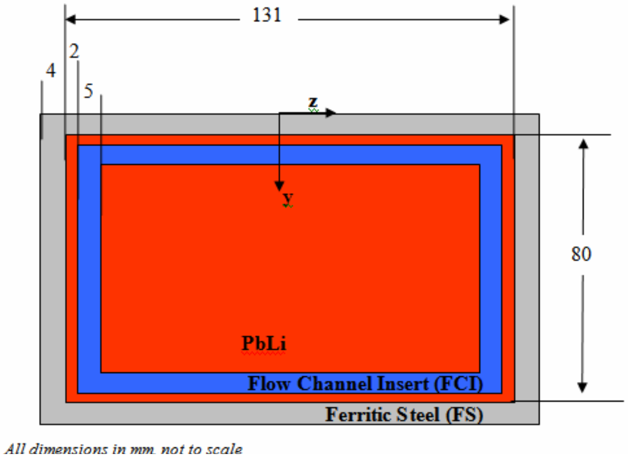


Figure 45: Schematic diagram showing cross-section of thermal blanket module.

The computations are currently underway, using the NERSC parallel supercomputer Bassi. However, some preliminary results are presented here. In both the y and z directions 118 grid points were used, with the nodes clustered such that they were close at the liquid wall and the outer boundaries. The Roberts stretching formula was used in the central Pb/Li core flow, and exponential stretching in the other regions. Uniform spacing of the nodes in the streamwise (x) direction was used.

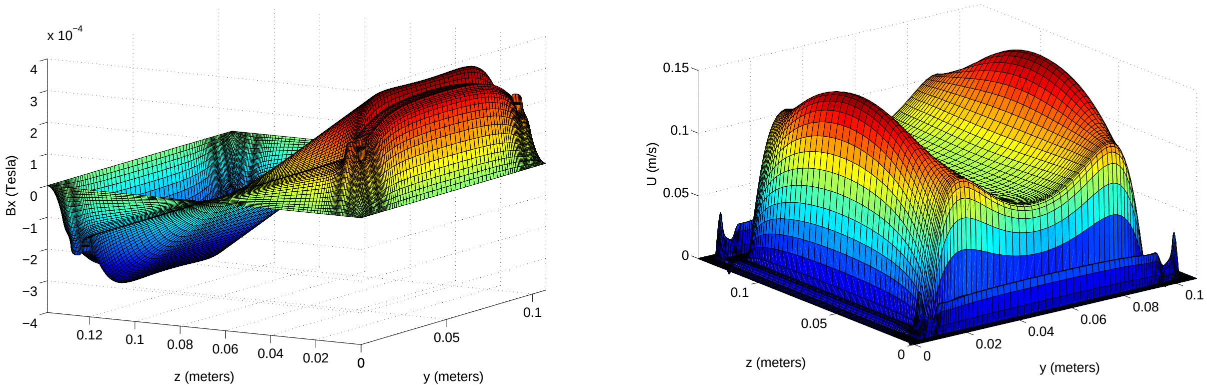


Figure 46: Induced field (Bx) and velocity (U) plots on cross-section of TBM module.

A pressure gradient was imposed to drive the flow. It was found that during the convergence process, the velocities fluctuated greatly though the amplitude of these fluctuation did decay rapidly. To avoid numerical problems, the pressure gradient was ramped up in stages. The reason for the fluctuations is not clear, but is thought to be associated with the preconditioning methdod used to accelerate convergence to steady state. Very similar behaviour was observed by Smolentsev et al. (2005); in both codes, these fluctuations were only observed when there were conducting walls. A Hartmann number of about 100 was used. Due to the dependence of the viscosity on temperature, the effective Ha increased along the length of the channel. In practice, higher Hartmann numbers would be used in the real situation, but here the objective was to obtain a similar flow distribution for the purposes of calculating the temperature profile, and value used here was high enough that the Hartmann and side layers were small compared with the central region. Naturally, higher Ha flows could be simulated with more grid points and the used of more processors with parallel jobs.

Figure 46 shows the induced field and the velocity profile on a cross-section of the cell. The velocity can be seen to have peaks near the side walls – this is due to the finite conductivity of the SiC composite in the FCI, and peaks would not be observed for a perfectly insulating FCI. In the other direction, there is a very steep climb in the velocity near the walls (the Hartmann layers). The velocity in the narrow channels is low compared with the main channel, and is higher in the sections perpendicular to the applied field. The induced magnetic field shows the usual steep climb at the outer walls and a constant slope over most of the rest of the domain. Note the presence of a step near $z = 0$, this corresponds to the FCI which has low electrical conductivity, inhibiting changes in the field.

Temperature profiles at x=0.7m and x=1.4m are shown in Fig. 47; these stations correspond to planes halfway down the channel from the inlet, and at the end of the channel. The profile taken on an $x - y$ plane through the centre is shown in Fig. 48. The temperatures are highest in the regions near the front wall, where the volumetric heating is greatest, and elevations in the temperature can also be seen to peak in the FCI. The two plots are broadly similar, though the the temperature in the central region is higher in the second plot than the first. In fact, the temperature in the core flow increases roughly linearly with distance from the inlet to reach about 440°C at the end. One can also observe dips toward the front and back of the core flow region – these locations correspond to the side wall jets where the flow is faster. It should be noted that these are preliminary results – subsequent runs on the supercomputer with slightly different parameters and higher resolution are expected to show somewhat lower peaks in the region around the FCI.

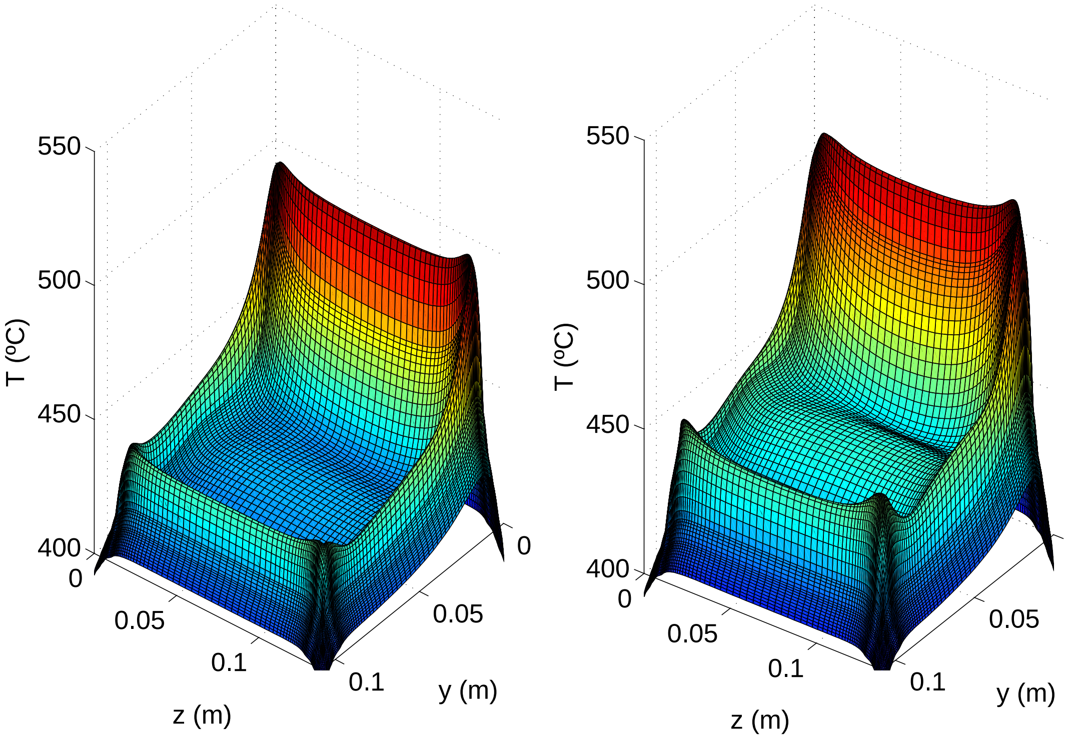


Figure 47: Temperature at 0.7m (left) and (1.4m right) from inlet.

These initial results with the MetaFlow code for the thermal blanket problem are encouraging.

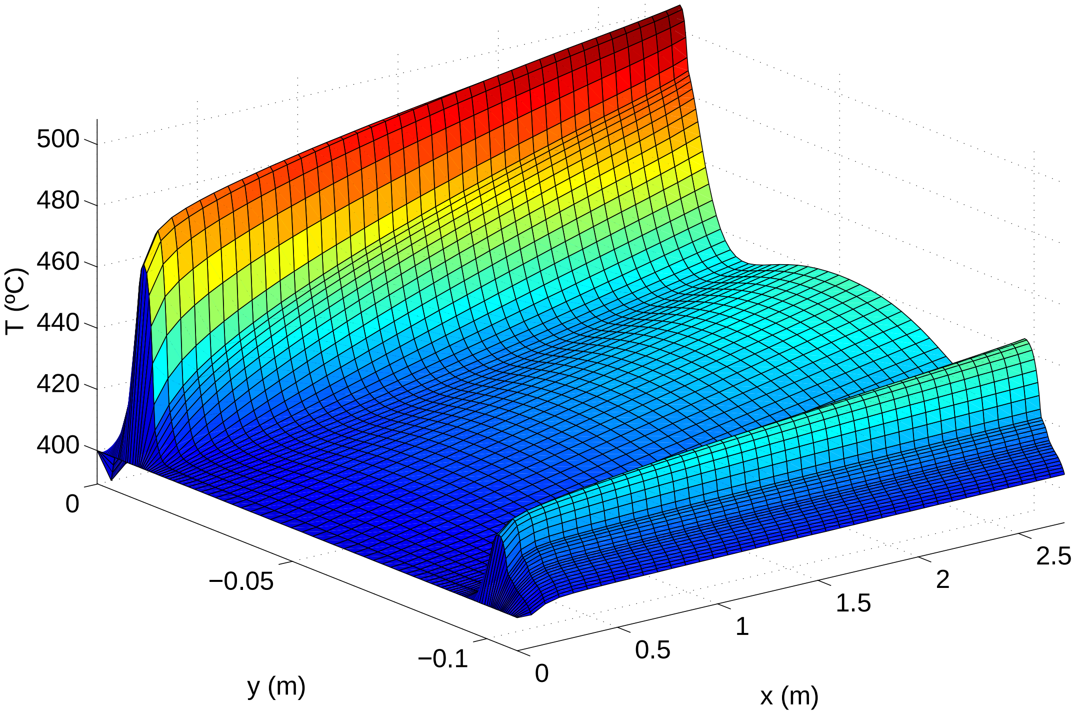


Figure 48: Temperature on $x - y$ plane through centre of duct.

9 Summary, Conclusions and Outlook

In this report, the development of three-dimensional CFD code for the simulation of complex flows has been documented. The core solver is based on the lattice Boltzmann method and uses the most advanced formulation available, the recently developed multiple relaxation time (MRT) model. This is supplemented by a lattice Boltzmann based models for MHD flows and a TVD scheme for heat transport. A key feature of all the numerical schemes used within MetaFlow is that they allow efficient parallelisation for modern supercomputers with distributed memory architectures.

The MRT implementation of the fluid solver has proven its accuracy in validation exercises involving a range of different flow scenarios. As well as a number of laminar cases, large eddy simulations (LES) of turbulent flows have been performed using a Smagorinsky model for the unresolved scales with wall damping. Excellent agreement with benchmark data has been found, including the challenging case of duct flow, where the secondary flows were successfully captured.

To put the present work in perspective, accuracy, stability and cost of MRT computations were compared with those of the single relaxation time (SRT) LBM – which is in more common use presently – with MRT showing significant advantages for eddy capturing simulations. For a given resolution, the MRT model was found to be more accurate and markedly more stable numerically – by at least a factor of 3 in Reynolds number for turbulent flow problems tested in this project, than the SRT-LBM. Also, after careful optimisation, the additional computational overhead of using the MRT in lieu of SRT-LBM was found to be small, between 10% and 30% depending on the machine tested. Thus, significantly higher Reynolds number simulations are possible than with the SRT-LBM. Moreover, the parallel performance tests carried out for up to 16 processors showed near linear scalability. In addition, the MRT implementation was found to be competitive when computational cost comparisons were made for a single processor with those of a finite-difference method (FDM) for LES. For the same case, and the same number of grid nodes employed and for the same number of time steps, the MRT was found to be faster than the FDM, by a factor of about 3 in a range of cases tested.

A key feature of MetaFlow is the incorporation of models for magnetohydrodynamic flows. As with the flow solver, the magnetic induction equation solver uses a lattice Boltzmann type approach. This fits well into the framework used for the flow calculations, in particular permitting efficient parallelisation. A number of extensions to the existing (and rather limited) LB models for MHD were developed. These advances allowed the application to wall-bounded flows and fluids with low magnetic Prandtl number, and a technique was developed to accelerate convergence of flows to steady-state. By use of appropriately stretched grids, flows with Hartmann numbers of several thousand could be efficiently simulated. Examples performed with MHD effect included duct flows, flow through a section of a thermal blanket and the simulation of turbulent flow through a pipe. Comparison of computed results with prior data, where available, showed good agreement.

The trend in computing is currently to be toward the use of supercomputers with very large numbers of processors. The lattice Boltzmann method is very well suited to this sort of computation, with almost linear scaling of computing speed with the number of processors. This is in contrast to other CFD techniques based on finite difference or finite volume, methods where the need to solve a Poisson-type pressure equation hampers the ability to efficiently parallelise. Also, the MRT model has proven to perform very well, and avoids the stability problems of earlier models which may have prevented wider acceptance of this method before. In view of this, lattice Boltzmann codes can be expected to become increasingly popular in the solution of very large computational problems in CFD and MHD.

10 References

Allain, J.P. et al. (2002) *FLIRE – flowing liquid surface retention experiment, design and testing.* Fusion Eng. and Des., **61–62**, 245–250

Ansumali, S., I.V. Karlin, and S. Succi (2004) *Kinetic theory of turbulence modeling: Smallness parameter, scaling and microscopic derivation of Smagorinsky model* Physica A **338**, 379–394

Arcidiacono, S. and 5 others (2006) *Simulation of binary mixtures with the lattice Boltzmann method.* Phys. Rev. E, **74**, 056707

Bird, R.B., W.E. Stewart and E.N. Lightfoot (2002) Transport Phenomena, 2nd Edition. John Wiley & Sons, New York.

Badalassi, V.E, H.D. Ceniceros, and S. Banerjee (2003) *Computation of multiphase systems with phase-field models*, J. Comput. Phys. **190** 371–397

Badalassi, V.E. (2005) *High-Order Semi-Implicit Schemes for Incompressible Flow Simulations*, to be submitted to J. Comput. Phys.

Bhatnagar, P., M. Gross and E. Krook (1954) *A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems* Physical Review **94**, 511–525

Breyiannis, G. and D. Valougeorgis (2004) *Lattice kinetic simulations in three-dimensional magnetohydrodynamics*. Phys. Rev. E, **69**, 65 702

Bouzidi, M., M. Firdaouss and P. Lallemand (2001) *Momentum transfer of a Boltzmann-lattice fluid with boundaries.* Phys. Fluids **13**, 3452-3459

Cao, N., Chen, S., Jin, S. and D. Martinez (1997) *Physical symmetry and lattice symmetry in the lattice Boltzmann method* Physical Review E **5**, R21-R24

Chapman, S. and T. Cowling (1964) *Mathematical Theory of Nonuniform Gases* Cambridge University Press, London.

Chen, H., S. Chen and W.H. Matthaeus (1992) *Recovery of the Navier–Stokes equations using a lattice-gas Boltzmann method.* Phys. Rev. A, **45**, 5339–5342

Chen, S., and G.D. Doolen (1998) *Lattice Boltzmann method for fluid flows.* Annu. Rev. Fluid Mech. **30**, 329–364

Chen S., D. Martinez and R. Mei (1996) *On boundary conditions in lattice Boltzmann methods.* Phys. Fluids **8**, 2527–2536

Chen, S., and G.D. Doolen (1998) *Lattice Boltzmann method for fluid flows.* Annu. Rev. Fluid Mech. **30**, 329–364

Cheesewright, J., G. McGrath and D. G. Petty (1990) Aeronautical Engineering Dept. Rep., ER 1011, Queen Mary Westfield College, London, UK

Chorin, A.J. (1968) *Numerical solution of the Navier–Stokes equations.* Mathematics of Computation **22**, 745–762

Deen, W.M. (1998) *Analysis of transport phenomena.* Oxford Univ. Press, New York

d'Humières, D., I. Ginzburg, M. Krafczyk, P. Lallemand and L.-S. Luo (2002) Multiple-relaxation-time lattice Boltzmann models in three-dimensions Phil. Trans. R. Soc. Lond. A **360**, pp. 437–451

Dellar, P.J. (2002) *Lattice kinetic schemes for magnetohydrodynamics.* J. Comp. Phys., **179** 95–126

Eggels, J.G.M., F. Unger, M.H. Weiss, J. Westerweel, R.J. Adrian, R. Friedrich and F.T.M. Niewstadt (1994) *Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment* J. Fluid Mech **268**, 175–209

Filippova, O. and D. Hänel (1998) *Grid refinement for Lattice–BGK models.* J. Comp. Phys. **147**, pp. 219–228

Gavrilakis S. (1992) *Numerical simulation of low-Reynolds number turbulent flow through a straight duct* J. Fluid Mech. **244**, pp. 101

Germano, M., Piomelli, U., Moin, P., and W.H Cabot (1991) *A dynamic subgrid-scale eddy viscosity model* Physics of Fluids A **3**, 1760–1765.

Ginzburg, I. and K. Steiner (2003) *Lattice Boltzmann model for free-surface flow and its application to filling process in casting.* J. Comp. Phys., **185**, 61-99

Grad, H. (1949) *On the kinetic theory of rarefied gases* Comm. Pure and App. Math. **2**, 331–407

Guo Z. and C. Zheng (2002) *An extrapolation method for boundary conditions in lattice Boltzmann method.* Phys. Fluids **14**, 2007-2010

Guo Z., T.S. Zhao and Y. Shi (2004) *Preconditioned lattice Boltzmann method for steady flows* Phys. Rev. E **70**, 066706

Harten, A., B. Engquist, S. Osher and S.R. Chakravarthy (1987) *Uniformly high order accurate essentially non-oscillatory schemes*, J. Comput. Phys. **71** 231–303

He, X., L.-S. Luo and M. Dembo (1996) *Some progress in the lattice Boltzmann method. Part 1: Non-uniform mesh grids.* J. Comp. Phys., **129**, 357–363

He, X. (1997) *Error analysis for the interpolation-supplemented lattice-Boltzmann equation scheme.* Int. J. Modern Phys. C , **8** 737–745

He, X., S. Chen and R. Zhang (1999) *A lattice Boltzmann scheme for incompressible multiphase flow and its application in Rayleigh-Taylor instability.* J. Comp. Phys. **152**, pp. 642–663

He, X., S. Chen and R. Zhang (1999) *A lattice Boltzmann scheme for incompressible multiphase flow and its application in Rayleigh-Taylor instability.* J. Comp. Phys. **152**, pp. 642–663

Higuera, F. and J. Himenez (1989) *Boltzmann approach to lattice gas simulations.* Europhysics Lett. **9**, 663–668

Higuera, F., S. Succi and R. Benzi (1989) *Lattice gas dynamics with enhanced collisions.* Europhysics Lett. **9**, 345–349

Huser, A. and S. Birigen (1993) *Direct numerical simulation of turbulent flow in a square duct* J. Fluid Mech. **257**, pp. 65

Junk, M. (2001) *A finite difference interpretation of the lattice Boltzmann method.* Numerical Methods for Partial Differential Equations **158**, 267-301.

Junk, M., Klar, A., and L.-S. Luo (2005) *Asymptotic analysis of the lattice Boltzmann equation.* Journal of Computational Physics **210**, 676-704.

Kim, J., Moin, P. and Moser, R. (1987) *Turbulence statistics in fully developed channel flow at low Reynolds number.* J. Fluid Mechanics **177**, 133-166.

Kreplin, H. and H. Eckelmann (1979) *Behavior of the three fluctuating velocity components in the wall region of a turbulent channel flow.* Physics of Fluids **28**, 1233

Ladd, A.J.C. (1994) *Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part I. Theoretical foundation.* J. Fluid Mechanics **271**, 285

Lallemand, P. and L.-S. Luo (2000) *Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability* Physical Review E **61**, 6546–6562

Lallemand P. and L.-S. Luo (2003a) *Lattice Boltzmann method for moving boundaries.* J. Comp. Phys. **184**, 406-421

Lallemand P. and L.-S. Luo (2003b) *Hybrid finite-difference thermal lattice Boltzmann equation.* Int. J. Mod. Phys. B **17**, 41-47

Lam, K. (1989) *Numerical investigation of turbulent flow*, PhD Thesis, Univ. California

Lam, K. and S. Banerjee (1992) *On the conditions of streak formation in bounded flows.* Phys. Fluids A **4**, 306 320.

Lammers, P., K.N. Beronov, R. Volkert, G. Brenner, and F. Durst (2006) *Lattice BGK direct numerical simulation of fully developed turbulence in incompressible plane channel flow.* Computers and Fluids, in press.

Lee, T. and C.-L. Lin (2005) *A stable discretization of the lattice-Boltzmann equation for simulation of incompressible two-phase flows at high density ratio.* J. Comp. Phys. **206**, pp. 16–47

Leveque, R.J. (1992) *Numerical Methods for Conservation Laws.* Birkhauser-Verlag, Basel.

Leveque, R.J. (2002) *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, New York.

Lu, Z., Y. Liao, D. Qian, J.B. McLaughlin, J.J. Derksen and K. Kontomaris (2002) *Large eddy simulation of a stirred tank using the lattice Boltzmann method on a nonuniform grid.* J. Comp Phys. **181**, pp. 675–704

Madabhushi, R.K. and S.P. Vanka (1991) *Large eddy simulation of turbulence-driven secondary flow in a square duct.* Phys. Fluids A **3**, pp. 2734

McCracken, M.E. and J. Abraham (2005) *Multiple-Relaxation-Time Lattice- Boltzmann Model for Multiphase Flow.* Physical Review E **71**, 036701.

Mei R., L.-S. Luo and W. Shyy (1999) *An accurate curved boundary treatment in the lattice Boltzmann method.* J. Comp. Phys. **155**, 307–330

Moin, P. and Kim, J. (1982) *Numerical investigation of turbulent channel flow.* J. Fluid Mechanics **118**, 341-377.

Morley, N.B., S. Smolentsev, R. Munipalli, M. Ni, D. Gao, M. Abdou (2004) *Progress on the modeling of liquid metal, free surface, MHD flows for fusion liquid walls.* Fusion Engineering and Design **72**, pp. 3-34

Moreau, R. (1990) *Magnetohydrodynamics.* Kluwer Academic Publications, Dordrecht, Netherlands

Müller, U. and L. Bühler (2001) *Magnetofluiddynamics in Channels and Containers.* Springer–Verlag, Berlin

Pan, Y. (1996) *Numerical Investigation of Particle-Containing Turbulent Open Channel Flows.* PhD Thesis, Dept. of Chemical Engineering, Univ. California, Santa Barbara

Pattison, M.J., K.N. Premnath and S. Banerjee (2007a) *Large eddy simulation of turbulent flow in a square duct using the generalized lattice Boltzmann equation.* Submitted to Phys. Rev. E

Pattison, M.J., K.N. Premnath, N.B Morley and M. Abdou (2007b) *Progress in lattice Boltzmann methods for magnetohydrodynamic flows relevant to fusion applications.* Submitted to Fusion Engineering & Design.

Premnath, K.N. and J. Abraham (2004) *Discrete lattice BGK Boltzmann equation computation of transient incompressible turbulent jet.* Int. J. Modern Phys. C **15**, 699–719

Premnath, K.N. and J. Abraham (2006) *Three dimensional multi-relaxation-time (MRT) lattice Boltzmann models for multiphase flow.* J. Comput. Phys. (in press)

Premnath, K.N., J. Abraham and V. Magi (2002) *Parallelization of a multidimensional code for the simulation of flows in engines: Performance with an OpenMP programming model.* Proc. High Performance Computing Symp., Advanced Simulation Tech. Conf. 2002, San Diego, CA

Premnath, K.N. (2006) *Generalised magnetic induction equation for spatially varying electrical conductivity.* MetaHeuristics Internal Report, Santa Barbara, CA

Premnath, K.N. and M.J. Pattison (2007) *Convergence acceleration of generalized lattice Boltzmann equation with forcing term through preconditioning for steady state flows and some applications.* Submitted to Physical Review E.

Premnath, K.N., M.J. Pattison and S. Banerjee (2007) *Generalized lattice Boltzmann equation with forcing term for LES of bounded turbulent flows: accuracy, stability and computational efficiency.* Submitted to Physical Review E.

Press, W.H., S.A. Teukolsky, W.T. Vetterling and B.P. Flannery (1997) *Numerical Recipes in C.* Cambridge University Press, Cantab., UK

Qian, Y.H, D. d'Humières and P. Lallemand (1992) *Lattice BGK models for the Navier-Stokes equation.* Europhys. Lett. **17**, pp. 479–484.

Roberts, G.O. (1971) *Computational meshes for boundary layer problems*, Proc. Second Int. Conf. Numerical Methods Fluid Dynamics, Lecture Notes on Physics **8**, 171–177

Salvetti, M.V. and S. Banerjee (1995) *A priori tests of a new dynamic subgrid-scale model for finite-difference large-eddy simulations* Physics of Fluids **7**, 2831–2847

Salvetti, M.V., Y. Zang, R.L. Street and S. Banerjee (1997) *Large-eddy simulations of free-surface decaying turbulence with dynamic subgrid-scale models* Physics of Fluids **9**, 2405–2419.

Sankaranarayanan, K., X. Shan, I.G. Kevrekidis and S. Sundaresan (2002) *Analysis of drag and virtual mass forces in bubbly solutions using an implicit formulation of the lattice Boltzmann method.* J. Fluid Mech. **452**, 61–96

Smolentsev, S., S. Cuevas and N.B. Morley (2004) *On the choice of variables in modeling liquid metal magnetohydrodynamic flows at low magnetic Reynolds number*, UCLA-FNT-202

Smolentsev, S., N.B. Morley and M.A. Abdou (2005) *Code development for analysis of MHD pressure drop in a liquid blanket using insulation technique based on a fully developed flow model.* Fusion Eng. and Des. **73**, 83–93

Smolentsev, S., N.B. Morley and M.A. Abdou (2006) *Magnetohydrodynamic and thermal issues on the* SiC_f*/SiC flow channel insert* Fusion Sci. and Tech. **50**, 107–119

Sone, Y. (1990) *Asymptotic theory of a steady flow of a rarefied gas past bodies of small Knudsen numbers* in: R. Gatignol, J.B. Soubbaramayer (Eds.) Proceedings of a Symposium Held in Honour of Henri Cabannes, Advances in Kinetic Theory and Continuum Mechanics, Springer, Berlin.

Succi, S. et al. (1999) *An integer lattice realization of a Lax scheme for transport processes in multiple component fluid flows.* J. Comp. Phys., **152**, 493–516

Succi, S. (2001) The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Clarendon Press, Oxford

Suratanakavikul, V. and A.J. Marquis (1999) *A comparative study of flux-limiters in unsteady and steady flows.* Proc. 13th National Mechanical Eng. Conf., 2–3 Dec. 1999, South Pattaya, Cholburi

Teman, R. (2001) *Navier-Stokes Equations: Theory and Numerical Analysis*, American Mathematical Society. (Original paper on a projection method by Teman appeared in: R. Teman (1968) Une méthode d'approximation de la solution des equations de Navier–Stokes, Bull. Soc. Math. France **98**, 115–152)

Toro, E.F. (1999) *Riemann solvers and numerical methods for fluid dynamics.* Springer-Verlag, Berlin.

Wolf-Gladrow, D. (2000) *Lattice-gas Cellular Automata and Lattice Boltzmann Models* Lecture Notes in Mathematics **1725**, Springer, New York.

Yu, D., R. Mei and W. Shyy (2002) *A multi-block lattice Boltzmann method for viscous fluid flows.* Int. J. Numer. Meth. Fluids, **39**, 99–120

Yu, D., R. Mei, L.-S. Luo and W. Shyy (2003) *Viscous flow computations with the method of lattice Boltzmann equation.* Progress in Aerospace Sciences **39**, 329–367

Yu, H.., L.-S. Luo and S. Girimaji (2006) *LES of turbulent square jet flow using an MRT lattice Boltzmann model* Computers and Fluids **35**, 957–965

Zhang, Y., Street, R.L. and J.R. Koseff (1993) *A dynamic subgrid-scale model and its application to turbulent recirculating flows* Physics of Fluids A **5** 3186

Zou, Q. and X. He (1997) *On pressure and velocity boundary conditions for the lattice Boltzmann BGK model.* Phys. Fluids **9**, 1591-1598

A Publications Arising from this Project

Pattison, Martin J., Kannan N. Premnath and Sanjoy Banerjee (2007) Turbulence-Induced Secondary Flows in a Square Duct using a Multiple-Relaxation-Time Lattice-Boltzmann Approach. Turbulence and Shear Flow Phenomena (TSFP)-5, 27-29 August 2007, Munich, Germany.

Pattison, Martin J., Kannan N. Premnath and Neil B. Morley (2006) Lattice Boltzmann Methods for Magnetohydrodynamic Flows in Fusion Applications. American Nuclear Society (ANS) Winter Meeting, 12-16 November 2006, Albuquerque, NM

Premnath, Kannan N. and Martin J. Pattison (2007) Convergence Acceleration of Generalized Lattice Boltzmann Equation with Forcing Term through Preconditioning for Steady State Flows and Some Applications. Submitted to Physical Review E.

Premnath, Kannan N., Martin J. Pattison and Sanjoy Banerjee (2007) Generalized Lattice Boltzmann Equation with Forcing Term for LES of Bounded Turbulent Flows: Accuracy, Stability and Computational Efficiency. Submitted to Physical Review E.

Pattison, Martin J., Kannan N. Premnath, Neil B. Morley and Mohamed A. Abdou (2007) Progress in Lattice Boltzmann Methods for Magnetohydrodynamic Flows relevant to Fusion Applications. Submitted to Fusion Engineering & Design.

Pattison, Martin J., Kannan N. Premnath and Sanjoy Banerjee (2006) Large Eddy Simulation of Turbulent Flow in a Square Duct using the Generalized Lattice Boltzmann Equation. Submitted to Phys. Rev. E

Pattison, Martin J. and Sanjoy Banerjee (2004) Numerical simulation of fluids using the lattice Boltzmann scheme. 6th Int. Conf. Nuclear Thermal Hydraulics, Operations and Safety (NUTHOS-6). Nara, Japan, 4-8 Oct. 2004

B Components of Moments for the D3Q19 Lattice

The components of the various elements in the moments are as follows d'Humières et al. (2002):

$$
\begin{array}{llllll}
\widehat{f_0} = \rho & & & & & \\
\widehat{f_1} = e & \widehat{f_2} = e^2 & \widehat{f_3} = j_x & \widehat{f_4} = q_x & \widehat{f_5} = j_y & \widehat{f_6} = q_y \\
\widehat{f_7} = j_z & \widehat{f_8} = q_z & \widehat{f_9} = 3p_{xx} & \widehat{f_{10}} = 3\pi_{xx} & \widehat{f_{11}} = p_{ww} & \widehat{f_{12}} = \pi_{ww} \\
\widehat{f_{13}} = p_{xy} & \widehat{f_{14}} = p_{yz} & \widehat{f_{15}} = p_{xx} & \widehat{f_{16}} = m_x & \widehat{f_{17}} = m_y & \widehat{f_{18}} = m_z
\end{array}
$$

In addition to the conserved hydrodynamic moments, *viz.*, density and components of momentum (ρ, j_x, j_y and j_z), we have the following non-conserved moments, some of which have direct physical import and are all relaxed during collision process: various components of the symmetric traceless stress tensor ($3p_{xx}$, p_{ww}, p_{xy}, and p_{yz}, where $p_{ww} = p_{yy} - p_{zz}$, and $\sum_{i=x,y,z} p_{ii} = 0$), components of heat flux, q_x, q_y and q_z, kinetic energy (e), square of kinetic energy (e^2), and the rest are based on certain combinations of polynomials of the momentum.

C Transformation Matrix for the D3Q19 Lattice

The transformation matrix $\mathcal{T}$ can be written as (d'Humières et al., 2002):

$$\mathcal{T} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-30 & -11 & -11 & -11 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
12 & -4 & -4 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & -4 & 4 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & -4 & -4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -2 & -2 & 2 & 2 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1
\end{bmatrix}$$

From this transformation matrix, a diagonal scaling matrix $\widehat{L}$, whose elements correspond to the sum of the inner products of the elements of the respective row of $\mathcal{T}$ can be constructed. This $\widehat{L}$, which is given in the following, is useful in actual implementation of the MRT as the transformation matrix $\mathcal{T}$ itself is never used directly in calculations in order to optimize the computational speed:

$$\widehat{\mathcal{L}} = diag\,(19, 2394, 252, 10, 40, 10, 40, 10, 40, 36, 72, 12, 24, 4, 4, 4, 8, 8, 8)$$

It may be noted that since $\mathcal{T}$ is row-wise orthogonal, its transpose $\mathcal{T}^t$ is related to its inverse $\mathcal{T}^{-1}$ through $\mathcal{T}^{-1} = \mathcal{T}^t\widehat{L}$. Thus, the inverse of the transformation matrix is never need to be carried out explicitly, but can always be substituted by the product of the transpose of the transformation matrix and the diagonal scaling matrix $\widehat{L}$. This scaling matrix will be useful in the optimization strategies discussed in Appendix G in this report.

D Equilibrium Distributions in Moment Space for the D3Q19 Lattice

The equilibrium distributions in moment space for each of the 19 discrete particle velocity directions for the D3Q19 lattice, which are functions of the conserved hydrodynamic fields, i.e. fluid density ρ and momentum, $j_k(=\rho u_k)$ are as follows (d'Humières et al., 2002):

$$\widehat{f}_0^{\mathrm{eq}} = \rho$$

$$\widehat{f}_1^{\mathrm{eq}} \equiv e^{eq} = -11\rho + 19\left(\frac{j_x^2+j_y^2+j_z^2}{\rho}\right) \qquad \widehat{f}_2^{\mathrm{eq}} \equiv e^{2,eq} = 3\rho - \frac{11}{2}\left(\frac{j_x^2+j_y^2+j_z^2}{\rho}\right) \qquad \widehat{f}_3^{\mathrm{eq}} = j_x$$

$$\widehat{f}_4^{\mathrm{eq}} \equiv q_x^{eq} = -\frac{2}{3}j_x \qquad \widehat{f}_5^{\mathrm{eq}} = j_y \qquad \widehat{f}_6^{\mathrm{eq}} \equiv q_y^{eq} = -\frac{2}{3}j_y$$

$$\widehat{f}_7^{\mathrm{eq}} = j_z \qquad \widehat{f}_8^{\mathrm{eq}} \equiv q_z^{eq} = -\frac{2}{3}j_z \qquad \widehat{f}_9^{\mathrm{eq}} \equiv 3p_{xx}^{eq} = \frac{[2j_x^2-(j_y^2+j_z^2)]}{\rho}$$

$$\widehat{f}_{10}^{\mathrm{eq}} \equiv 3\pi_{xx}^{eq} = 3\left(-\frac{1}{2}p_{xx}^{eq}\right) \qquad \widehat{f}_{11}^{\mathrm{eq}} \equiv p_{ww}^{eq} = \frac{[j_y^2-j_z^2]}{\rho} \qquad \widehat{f}_{12}^{\mathrm{eq}} \equiv \pi_{ww}^{eq} = -\frac{1}{2}p_{ww}^{eq}$$

$$\widehat{f}_{13}^{\mathrm{eq}} \equiv p_{xy}^{eq} = \frac{j_xj_y}{\rho} \qquad \widehat{f}_{14}^{\mathrm{eq}} \equiv p_{yz}^{eq} = \frac{j_yj_z}{\rho} \qquad \widehat{f}_{15}^{\mathrm{eq}} \equiv p_{xz}^{eq} = \frac{j_xj_z}{\rho}$$

$$\widehat{f}_{16}^{\mathrm{eq}} \equiv m_x^{eq} = 0 \qquad \widehat{f}_{17}^{\mathrm{eq}} \equiv m_y^{eq} = 0 \qquad \widehat{f}_{18}^{\mathrm{eq}} \equiv m_z^{eq} = 0$$

E Source Terms in Moment Space for the D3Q19 Lattice

Earlier work (Premnath and Abraham, 2006) derived the source terms explicitly in terms of force field for the D3Q15 lattice. In this project we are using the D3Q19 lattice which was found to have greater numerical stability characteristics for simulating turbulent flows than that with the D3Q15 lattice. We derived the expressions for the source terms $\widehat{S}_\alpha$ representing the external force field $\boldsymbol{F} = (F_x, F_y, F_z)$ imposed on the fluid in moment space explicitly and only summarize the results below for brevity:

$$\begin{aligned} S_\alpha &= \omega_\alpha \left[\frac{3}{c^2} \left(e_{\alpha x} - u_x \right) + \frac{9}{c^4} \left(\overrightarrow{e_\alpha} \cdot \overrightarrow{u} \right) e_{\alpha x} \right] \\ &+ \omega_\alpha \left[\frac{3}{c^2} \left(e_{\alpha y} - u_y \right) + \frac{9}{c^4} \left(\overrightarrow{e_\alpha} \cdot \overrightarrow{u} \right) e_{\alpha y} \right] + \omega_\alpha \left[\frac{3}{c^2} \left(e_{\alpha z} - u_z \right) + \frac{9}{c^4} \left(\overrightarrow{e_\alpha} \cdot \overrightarrow{u} \right) e_{\alpha z} \right] \end{aligned} \tag{E–1}$$

$$\mathbf{S} = \left[S_0, S_1, S_2, \ldots, S_{18} \right]^t \tag{E–2}$$

$$\widehat{\mathbf{S}} = \left[\widehat{S}_0, \widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{18} \right]^t \tag{E–3}$$

$$\widehat{\mathbf{S}} = \mathcal{T} \mathbf{S} \tag{E–4}$$

$$\begin{array}{lll} \widehat{S}_0 = 0 & & \\ \widehat{S}_1 = 38(F_x u_x + F_y u_y + F_z u_z) & \widehat{S}_2 = -11(F_x u_x + F_y u_y + F_z u_z) & \widehat{S}_3 = F_x \\ \widehat{S}_4 = -\frac{2}{3} F_x & \widehat{S}_5 = F_y & \widehat{S}_6 = -\frac{2}{3} F_y \\ \widehat{S}_7 = F_z & \widehat{S}_8 = -\frac{2}{3} F_z & \widehat{S}_9 = 2(2F_x u_x - F_y u_y - F_z u_z) \\ \widehat{S}_{10} = -(2F_x u_x - F_y u_y - F_z u_z) & \widehat{S}_{11} = 2(F_y u_y - F_z u_z) & \widehat{S}_{12} = -(F_y u_y - F_z u_z) \\ \widehat{S}_{13} = (F_x u_y + F_y u_x) & \widehat{S}_{14} = (F_y u_z + F_z u_y) & \widehat{S}_{15} = (F_x u_z + F_z u_x) \\ \widehat{S}_{16} = 0 & \widehat{S}_{17} = 0 & \widehat{S}_{18} = 0 \end{array}$$

F Optimization of Computational Procedure for Solution of the MRT–LBM with Forcing Term

Before we discuss the optimized computational procedure for solution of the MRT–LBM, some useful expressions are developed in this regard. For brevity in presentation, define the following column vectors representing distribution functions, their local equilibrium values and corresponding source terms, respectively, for the D3Q19 lattice:

$$\mathbf{f} = \left[f_0, f_1, f_2, \ldots, f_{18} \right]^t \tag{F–1}$$

$$\mathbf{f}^{eq} = \left[f_0^{eq}, f_1^{eq}, f_2^{eq}, \ldots, f_{18}^{eq} \right]^t \tag{F–2}$$

$$\mathbf{S}^{eq} = \left[S_0^{eq}, S_1^{eq}, S_2^{eq}, \ldots, S_{18}^{eq} \right]^t \tag{F–3}$$

Also, define the Cartesian components of the particle velocity directions and spatial location:

$$\mathbf{e}_i = \left[e_{0i}, e_{1i}, e_{2i}, \ldots, e_{18i} \right]^t, \qquad i \in \{x, y, z\}, \qquad \overrightarrow{x} = (x, y, z) \tag{F–4}$$

Thus the effect of particle collisions, which provides amongst other things the viscous behavior of the fluid, and the imposed forces through the source terms can be represented by the "effective" collision column vector (see Eq. (1) in the main text)

$$\boldsymbol{\varpi}_{(\overrightarrow{x},t)} = -\Lambda \left(\mathbf{f} - \mathbf{f}^{eq} \right)_{(\overrightarrow{x},t)} + \left(\mathcal{I} - \frac{1}{2} \Lambda \right) \mathbf{S}_{(\overrightarrow{x},t)} \tag{F–5}$$

So the column vector representing the post-collision distribution functions may be written as (see Eq. 7–a in the main text)

$$\widetilde{\mathbf{f}}_{(\overrightarrow{x},t)} = \mathbf{f}_{(\overrightarrow{x},t)} + \boldsymbol{\varpi}_{(\overrightarrow{x},t)} \tag{F–6}$$

and the streaming step can be represented through

$$\mathbf{f}_{(\overrightarrow{x}+\overrightarrow{\delta_s}\delta_t,t+\delta_t)} = \widetilde{\mathbf{f}}_{(\overrightarrow{x},t)}, \qquad \overrightarrow{\delta_s} = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z) \tag{F–7}$$

Transform the column vectors given in particle velocity space to moment space, defined by the "hat" symbol, through the transformation matrix $\mathcal{T}$ given in Appendix C

$$\widehat{\mathbf{f}} = \mathcal{T}\mathbf{f} \qquad \widehat{\mathbf{f}}^{eq} = \mathcal{T}\mathbf{f}^{eq} \qquad \widehat{\mathbf{S}} = \mathcal{T}\mathbf{S} \tag{F–8}$$

Since $\widehat{\Lambda} = \mathcal{T}\Lambda\mathcal{T}^{-1}$, $\boldsymbol{\varpi}_{(\overrightarrow{x},t)}$ in Eq. (F–5) can be written in moment space as

$$\boldsymbol{\varpi}_{(\overrightarrow{x},t)} = \mathcal{T}^{-1}\left[-\widehat{\Lambda}\left(\widehat{\mathbf{f}} - \widehat{\mathbf{f}}^{eq}\right)_{(\overrightarrow{x},t)} + \left(\mathcal{I} - \frac{1}{2}\widehat{\Lambda}\right)\widehat{\mathbf{S}}_{(\overrightarrow{x},t)}\right] \tag{F–9}$$

Using the row-wise orthogonal property of the transformation matrix, we can write the inverse in terms of the transpose

$$\mathcal{T}^{-1} = \mathcal{T}^t\widehat{\mathcal{L}} \tag{F–10}$$

$$\boldsymbol{\varpi}_{(\overrightarrow{x},t)} = \mathcal{T}^t\widehat{\boldsymbol{q}} \tag{F–11}$$

where

$$\widehat{\mathbf{q}} = \left[-\widehat{\mathcal{L}}^{-1}\widehat{\Lambda}\left(\widehat{\boldsymbol{f}} - \widehat{\boldsymbol{f}}^{eq}\right)_{(\overrightarrow{x},t)} + \left(\widehat{\mathcal{L}}^{-1} - \frac{1}{2}\widehat{\mathcal{L}}^{-1}\widehat{\Lambda}\right)\widehat{\boldsymbol{S}}_{(\overrightarrow{x},t)}\right] \tag{F–12}$$

To simplify notation, let us define a "scaled" diagonal relaxation matrix in moment space

$$\widehat{\Gamma} = \widehat{\mathcal{L}}^{-1}\widehat{\Lambda} \tag{F–13}$$

which may be written by substituting for $\widehat{L}^{-1}$ from Appendix C.

$$\begin{aligned}\widehat{\Gamma} = diag\Big(&\frac{1}{19}s_0, \frac{1}{2394}s_1, \frac{1}{252}s_2, \frac{1}{10}s_3, \frac{1}{40}s_4, \frac{1}{10}s_5, \frac{1}{40}s_6, \frac{1}{10}s_7, \frac{1}{40}s_8, \frac{1}{36}s_9, \frac{1}{72}s_{10},\\ &\frac{1}{12}s_{11}, \frac{1}{24}s_{12}, \frac{1}{4}s_{13}, \frac{1}{4}s_{14}, \frac{1}{4}s_{15}, \frac{1}{8}s_{16}, \frac{1}{8}s_{17}, \frac{1}{8}s_{18}\Big)\end{aligned} \tag{F–14}$$

Thus, Eq. (F–12) can be rewritten as

$$\widehat{\mathbf{q}} = \left[-\widehat{\Gamma}\left(\widehat{\mathbf{f}} - \widehat{\mathbf{f}}^{eq}\right)_{(\overrightarrow{x},t)} + \left(\widehat{\mathcal{L}}^{-1} - \frac{1}{2}\widehat{\Gamma}\right)\widehat{\mathbf{S}}_{(\overrightarrow{x},t)}\right] \tag{F–15}$$

Equations (F–11), (F–14) and (F–15), which are in moment space, are computationally more efficient than Eq. (F–5), which is in *particle velocity space*, for the effective collision step.

Now, from Eqs. (13) and (12) in the main text, some of the relaxation parameters of the relaxation matrix $\widehat{\Lambda} = diag(s_0, s_1, \ldots, s_{18})$ can be related to the transport coefficients and modulated by eddy viscosity as follows:

$$s_0^{-1} = s_\zeta^{-1} = \frac{9}{2}\zeta + \frac{1}{2} \tag{F–16}$$

$$s_9 = s_{11} = s_{13} = s_{14} = s_{15} = s_\nu \qquad \text{(F–17)}$$

where

$$s_\nu^{-1} = 3\nu + \frac{1}{2} = 3(\nu_0 + \nu_t) + \frac{1}{2} \qquad \text{(F–18)}$$

Here, eddy viscosity ν_t is given by the standard Smagorinsky SGS model modified by the van Driest wall damping as discussed in section 3.3 in the main text. The strain rate tensors required in the SGS model are computed directly from the non-equilibrium part of moments given in Appendix G. ν_0 is the molecular kinematic viscosity and is specified from the statement of the problem through dimensional group such as Reynolds number.

Other parameters can be tuned to enhance numerical stability at higher Reynolds numbers for a given lattice grid resolution. Based on the von Neumann stability analysis of the linearized MRT–LBM, the following values for the other relaxation parameters are determined (d'Humières et al., 2002): $s_1 = 1.19$, $s_2 = s_{10} = s_{12} = 1.4$, $s_4 = s_6 = s_8 = 1.2$, and $s_{16} = s_{17} = s_{18} = 1.98$. For the conserved moments, the values of the relaxation parameters are immaterial as their corresponding equilibrium distribution is set to the value of the respective moments itself. We could set $s_0 = s_3 = s_5 = s_7 = 1$ for simplicity. It may be noted that all relaxation parameter have the following bounds $0 < s_\alpha < 2$. In this project we use these values for the parameters.

With this prelude, the procedure to perform LES through an *optimized* MRT is as follows:

For every lattice node representing fluid region, at time step t

1. Compute the 19 elements of the forcing term column vector in *moment* space, $\widehat{\mathbf{S}}$. (This is explicitly given in terms of imposed force field $\overrightarrow{F} = (F_x, F_y, F_z)$ and velocity field $\overrightarrow{u} = (u_x, u_y, u_z)$ in Appendix E).

2. Transform distribution function in *particle velocity* space to *moment* space through $\widehat{\mathbf{f}} = \mathcal{T}\mathbf{f}$. Here, explicit matrix multiplication must be avoided as matrix $\mathcal{T}$ contains several zeroes and common sub expressions, involving only integers. In other words, express each of the 19 elements of $\widehat{\mathbf{f}}$ directly by the expanded form as a group of terms involving the elements of $\mathbf{f}$ (most compact version).

3. Compute the 19 elements of the equilibrium distributions in *moment* space $\widehat{\mathbf{f}}^{eq}$ which are given in terms of the algebraic expressions of the conserved hydrodynamic fields, i.e. local fluid density and momentum (see Appendix D).

4. Obtain the 19 elements of the diagonal relaxation matrix in *moment* space $\widehat{\Lambda}$ modified by the Smagorinsky SGS model. See Eqs. F–16 and F–17 and discussion therein.

5. Obtain the 19 elements of the "scaled" diagonal relaxation matrix $\widehat{\Gamma}$ from Eq. (F–14).

6. Compute the intermediate column vector $\widehat{\mathbf{q}}$ in Eq. (F–15).

7. Perform the "effective" collision operation by updating the column vector ϖ in *particle velocity* space in Eq. (F–11). As in Step 2, the matrix multiplication in $\varpi_{(\overrightarrow{x},t)} = \mathcal{T}^t\widehat{\mathbf{q}}$ must not be carried out explicitly. Since $\mathcal{T}^t$ is just the transpose of $\mathcal{T}$, all the possible properties pertaining to $\mathcal{T}$ stated in Step 2 must be exploited. Thus, express each of the 19 elements of ϖdirectly by the expanded form as a group of the most compact terms involving the elements of $\widehat{\mathbf{q}}$.

8. Compute the 19 elements of the post-collision distribution function in *particle velocity* space $\widetilde{\mathbf{f}}$ through Eq. (F–6).

9. Perform "streaming" step in *particle velocity* space through Eq. (F–7). This updates the 19 elements of the distribution functions at $\mathbf{f}$ for the time step $t + \delta_t$.

10. In the streaming step given in Step 9, if for particle velocity directions $\overrightarrow{e_{\bar{\alpha}}}$, where $\overrightarrow{e_{\bar{\alpha}}} = -\overrightarrow{e_\alpha}$, the incoming distribution function is from a boundary node, apply appropriate boundary conditions.

11. Compute hydrodynamic fields ρ, $\overrightarrow{j}\,(=\rho\overrightarrow{u})$ and p from the updated distribution functions through Eqs. 8–10. These quantities represent the grid-filtered fields as SGS effects are modeled by means of eddy viscosity through effective relaxation times in the collision step.

12. Repeat steps 1 - 12.

G Local Computation of Strain Rate Tensors through Non-equilibrium Moments in the MRT–LBM with Forcing Term

In this section, we derive a procedure for calculating the strain rate tensor from the MRT–LBM with forcing terms directly through the non-equilibrium part of the moments. They can be obtained by using the Chapman–Enskog analysis (Chapman and Cowling, 1964; Premnath and Abraham, 2006).

The non-equilibrium part of the distribution function may be written as

$$\widehat{\mathbf{f}} - \widehat{\mathbf{f}}^{eq} \approx \widehat{\mathbf{f}}^{(1)}\delta_t \tag{G–1}$$

First order expansion yields

$$\widehat{\mathbf{f}}^{(1)} = -\widehat{\Lambda}^{-1}\left(\partial_{t_0} + \widehat{\Xi}_1\partial_i\right)\widehat{\mathbf{f}}^{eq} + \left(\widehat{\Lambda} - \frac{1}{2}\right)\widehat{\mathbf{S}} \tag{G–2}$$

where

$$\mathbb{D}_t = diag\left(\partial_{t_0}, \partial_{t_0} + \overrightarrow{e_1}\cdot\overrightarrow{\nabla}, \ldots, \partial_{t_0} + \overrightarrow{e_{18}}\cdot\overrightarrow{\nabla}\right) \tag{G–3}$$

$$\widehat{\Xi}_{\alpha i} = \mathcal{T} e_{\alpha i}\mathcal{T}^{-1} \tag{G–4}$$

$$\widehat{\mathbb{D}_t} = \mathcal{T}\mathbb{D}_t\mathcal{T}^{-1} = diag\left(\partial_{t_0}, \partial_{t_0} + \widehat{\Xi}_{1i}\partial_i, \ldots, \partial_{t_0} + \widehat{\Xi}_{18i}\partial_i\right) \tag{G–5}$$

The above equation for $\widehat{\mathbf{f}}^{(1)}$ can be simplified considerably by noting that

$$\partial_{t_0}\left(\frac{j_x^2}{\rho}\right) \simeq 2u_xF_x, \quad \partial_{t_0}\left(\frac{j_y^2}{\rho}\right) \simeq 2u_yF_y, \quad \partial_{t_0}\left(\frac{j_z^2}{\rho}\right) \simeq 2u_zF_z, \tag{G–6}$$

$$\partial_{t_0}\left(\frac{j_xj_y}{\rho}\right) \simeq u_xF_y + u_yF_x, \quad \partial_{t_0}\left(\frac{j_yj_z}{\rho}\right) \simeq u_yF_z + u_zF_y, \quad \partial_{t_0}\left(\frac{j_xj_z}{\rho}\right) \simeq u_xF_z + u_zF_x, \tag{G–7}$$

$$\partial_{t_0}\rho = -\partial_k j_k, \tag{G–8}$$

Thus the non-equilibrium part of moments related to viscous stresses may be written as

$$\widehat{f}_1^{(1)} \equiv e^{(1)} = -\frac{1}{s_1}\left\{\partial_{t_0}\left(-11\rho + 19\frac{j_kj_k}{\rho}\right) + \frac{5}{3}\partial_k j_k\right\} + \left(\frac{1}{s_1} - \frac{1}{2}\right)\widehat{S}_1 \tag{G–9}$$

$$\widehat{f}_9^{(1)} \equiv 3p_{xx}^{(1)} = -\frac{1}{s_9}\left\{\partial_{t_0}\left(\frac{j_y^2 - j_z^2}{\rho}\right) + \frac{2}{3}\left(2\partial_x j_x - \partial_y j_y - \partial_z j_z\right)\right\} + \left(\frac{1}{s_9} - \frac{1}{2}\right)\widehat{S}_9 \tag{G–10}$$

$$\widehat{f}_{11}^{(1)} \equiv p_{ww}^{(1)} = -\frac{1}{s_{11}}\left\{\partial_{t_0}\left(\frac{j_y^2 - j_z^2}{\rho}\right) + \frac{2}{3}\left(\partial_y j_y - \partial_z j_z\right)\right\} + \left(\frac{1}{s_{11}} - \frac{1}{2}\right)\widehat{S}_{11} \tag{G–11}$$

$$\widehat{f}_{13}^{(1)} \equiv p_{xy}^{(1)} = -\frac{1}{s_{13}}\left\{\partial_{t_0}\left(\frac{j_x j_y}{\rho}\right) + \frac{1}{3}\left(\partial_x j_y + \partial_y j_x\right)\right\} + \left(\frac{1}{s_{13}} - \frac{1}{2}\right)\widehat{S}_{13} \tag{G–12}$$

$$\widehat{f}_{14}^{(1)} \equiv p_{yz}^{(1)} = -\frac{1}{s_{14}}\left\{\partial_{t_0}\left(\frac{j_y j_z}{\rho}\right) + \frac{1}{3}\left(\partial_y j_z + \partial_z j_y\right)\right\} + \left(\frac{1}{s_{14}} - \frac{1}{2}\right)\widehat{S}_{14} \tag{G–13}$$

$$\widehat{f}_{15}^{(1)} \equiv p_{xz}^{(1)} = -\frac{1}{s_{15}}\left\{\partial_{t_0}\left(\frac{j_x j_z}{\rho}\right) + \frac{1}{3}\left(\partial_x j_z + \partial_z j_x\right)\right\} + \left(\frac{1}{s_{15}} - \frac{1}{2}\right)\widehat{S}_{15} \tag{G–14}$$

Simplifying this further, we get

$$\widehat{f}_1^{(1)} = -\frac{38}{3}\frac{1}{s_1}\partial_k j_k - \frac{1}{2}\widehat{S}_1 \tag{G–15}$$

$$\widehat{f}_9^{(1)} = -\frac{2}{3}\frac{1}{s_9}\left(3\partial_x j_x - \partial_k j_k\right) - \frac{1}{2}\widehat{S}_9 \tag{G–16}$$

$$\widehat{f}_{11}^{(1)} = -\frac{2}{3}\frac{1}{s_{11}}\left(3\partial_y j_y - \partial_z j_z\right) - \frac{1}{2}\widehat{S}_{11} \tag{G–17}$$

$$\widehat{f}_{13}^{(1)} = -\frac{1}{3}\frac{1}{s_{13}}\left(\partial_x j_y + \partial_y j_x\right) - \frac{1}{2}\widehat{S}_{13} \tag{G–18}$$

$$\widehat{f}_{14}^{(1)} = -\frac{1}{3}\frac{1}{s_{14}}\left(\partial_y j_z + \partial_z j_y\right) - \frac{1}{2}\widehat{S}_{14} \tag{G–19}$$

$$\widehat{f}_{15}^{(1)} = -\frac{1}{3}\frac{1}{s_{15}}\left(\partial_x j_z + \partial_z j_x\right) - \frac{1}{2}\widehat{S}_{15} \tag{G–20}$$

By solving the above set of equations for components of various velocity gradients or strain rate tensors, we get

$$\begin{aligned}
S_{xx} &= -\frac{1}{38\rho}\left[s_1\widehat{h}_1^{(neq)} + 19 s_9\widehat{h}_9^{(neq)}\right] && \text{(G–21)}\\
S_{yy} &= -\frac{1}{76\rho}\left[2s_1\widehat{h}_1^{(neq)} - 19\left(s_9\widehat{h}_9^{(neq)} - 3s_{11}\widehat{h}_{11}^{(neq)}\right)\right] && \text{(G–22)}\\
S_{zz} &= -\frac{1}{76\rho}\left[2s_1\widehat{h}_1^{(neq)} - 19\left(s_9\widehat{h}_9^{(neq)} + 3s_{11}\widehat{h}_{11}^{(neq)}\right)\right] && \text{(G–23)}\\
S_{xy} &= -\frac{3}{2\rho}s_{13}\widehat{h}_{13}^{(neq)} && \text{(G–24)}\\
S_{yz} &= -\frac{3}{2\rho}s_{14}\widehat{h}_{14}^{(neq)} && \text{(G–25)}\\
S_{xz} &= -\frac{3}{2\rho}s_{15}\widehat{h}_{15}^{(neq)} && \text{(G–26)}
\end{aligned}$$

where $\widehat{h}_\alpha^{(neq)}$ is the non-equilibrium part of the moments supplemented by source terms (representing forces, see Appendix E).

$$\widehat{h}_\alpha^{(neq)} = \widehat{f}_\alpha - \widehat{f}_\alpha^{eq} + \frac{1}{2}\widehat{S}_\alpha, \qquad \alpha \in \{1, 9, 11, 13, 14, 15\} \tag{G–27}$$

Expressions for $\widehat{f}_\alpha$, $\widehat{f}_\alpha^{(eq)}$ and $\widehat{S}_\alpha$ are given in Appendices B, D and E respectively. From these expressions the eddy viscosity, ν_τ, needed in the Smagorinsky SGS model (Smagorinsky, 1963) can be calculated as follows:

$$\nu_t = (C_s \Delta)^2 \, |S| \tag{G–28}$$

where

$$\begin{aligned} |S| &= \sqrt{2 S_{ij} S_{ij}} && \text{(G–29)} \\ &= \sqrt{2(S_{xx}^2 + S_{yy}^2 + S_{zz}^2 + 2(S_{xy}^2 + S_{yz}^2 + S_{xz}^2))} && \text{(G–30)} \end{aligned}$$

The use of Eqs. (G–21–G–30) circumvents the usual finite difference calculation of the derivatives of velocity fields, i.e. $S_{ij} = \frac{1}{2}\left(\partial_j u_i + \partial_i u_j\right)$, and thus presents a completely local procedure. As a result, it becomes computationally efficient for complex geometries. The expressions for calculating strain rate tensor presented here generalises those presented in a recent work by Yu et al. (2006) for the case of MRT-LBM with forcing term.

H Accelerating Steady-State Convergence of Flow and Magnetic Fields through Preconditioning

For the case of steady state flow, the equation solved is:

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \frac{1}{\gamma} \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \boldsymbol{u}) = -\frac{1}{\gamma} \boldsymbol{\nabla} p^* + \frac{1}{\gamma} \boldsymbol{\nabla} \cdot (\rho \nu \boldsymbol{\nabla} \boldsymbol{u}) + \frac{\boldsymbol{F}}{\gamma} \tag{H–1}$$

where the pre-conditioning parameter γ controls the acceleration of convergence.

The preconditioned Navier–Stokes equations can be recovered by solving the pre-conditioned MRT–LBM

$$\widehat{\mathbf{f}}(\boldsymbol{x} + \boldsymbol{e}_\alpha \delta t, t + \delta t) - \widehat{\mathbf{f}}(\boldsymbol{x}, t) = -\mathcal{T} \widehat{\Lambda}' \left(\widehat{\mathbf{f}} - \widehat{\mathbf{f}}^{eq,P} \right) + \mathcal{T} (\mathbf{I} - \frac{1}{2} \widehat{\boldsymbol{\Lambda}}') \widehat{\boldsymbol{P}_s} \widehat{\mathbb{S}} \delta t \tag{H–2}$$

where the terms for $\widehat{\boldsymbol{f}}^{eq}$, $\widehat{\boldsymbol{\Lambda}'}$ and $\widehat{\boldsymbol{P}_s}$ become

$$\begin{array}{llll}
\widehat{f}_0^{\text{eq,P}} = \rho & \widehat{f}_1^{\text{eq,P}} = -11\rho + 19\left(\frac{j_x^2 + j_y^2 + j_z^2}{\gamma \rho}\right) & \widehat{f}_2^{\text{eq,P}} = 3\rho - \frac{11}{2}\left(\frac{j_x^2 + j_y^2 + j_z^2}{\gamma \rho}\right) & \widehat{f}_3^{\text{eq,P}} = j_x \\
\widehat{f}_4^{\text{eq,P}} = -\frac{2}{3} j_x & \widehat{f}_5^{\text{eq,P}} = j_y & \widehat{f}_6^{\text{eq,P}} = -\frac{2}{3} j_y & \widehat{f}_7^{\text{eq,P}} = j_z \\
\widehat{f}_8^{\text{eq,P}} = -\frac{2}{3} j_z & \widehat{f}_9^{\text{eq,P}} = \frac{[2j_x^2 - (j_y^2 + j_z^2)]}{\gamma \rho} & \widehat{f}_{10}^{\text{eq,P}} = \omega_{xx} \frac{[2j_x^2 - (j_y^2 + j_z^2)]}{\gamma \rho} & \widehat{f}_{11}^{\text{eq,P}} = \frac{[j_y^2 - j_z^2]}{\gamma \rho} \\
\widehat{f}_{12}^{\text{eq,P}} = \omega_{xx} \frac{[j_y^2 - j_z^2]}{\gamma \rho} & \widehat{f}_{13}^{\text{eq,P}} = \frac{j_x j_y}{\gamma \rho} & \widehat{f}_{14}^{\text{eq,P}} = \frac{j_y j_z}{\gamma \rho} & \widehat{f}_{15}^{\text{eq,P}} = \frac{j_x j_z}{\gamma \rho} \\
\widehat{f}_{16}^{\text{eq,P}} = 0 & \widehat{f}_{17}^{\text{eq,P}} = 0 & \widehat{f}_{18}^{\text{eq,P}} = 0 &
\end{array}$$

The relaxation matrix alters due to preconditioning and can written as:

$$\widehat{\boldsymbol{\Lambda}}' = diag\big[s_0',\, s_1',\, s_2',\, s_3',\, s_4',\, s_4',\, s_6',\, s_7',\, s_8',\, s_9',\, s_{10}', s_{11}', s_{12}', s_{13}', s_{14}', s_{15}', s_{16}', s_{17}', s_{18}'\big]$$

The scaling matrix $\widehat{\boldsymbol{P}}_s$ for the source terms becomes:

$$\widehat{\boldsymbol{P}}_s = diag\big[0, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \frac{1}{\gamma}, \frac{1}{\gamma}, \frac{1}{\gamma}, \frac{1}{\gamma}, \frac{1}{\gamma}, \frac{1}{\gamma}, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, 0, 0, 0\big]$$

where the components or source terms are given by

$$\begin{array}{lll}
\widehat{S}_0 = 0 & & \\
\widehat{S}_1 = 38(F_x u_x + F_y u_y + F_z u_z) & \widehat{S}_2 = -11(F_x u_x + F_y u_y + F_z u_z) & \widehat{S}_3 = F_x \\
\widehat{S}_4 = -\frac{2}{3}F_x & \widehat{S}_5 = F_y & \widehat{S}_6 = -\frac{2}{3}F_y \\
\widehat{S}_7 = F_z & \widehat{S}_8 = -\frac{2}{3}F_z & \widehat{S}_9 = 2(2F_x u_x - F_y u_y - F_z u_z) \\
\widehat{S}_{10} = -(2F_x u_x - F_y u_y - F_z u_z) & \widehat{S}_{11} = 2(F_y u_y - F_z u_z) & \widehat{S}_{12} = -(F_y u_y - F_z u_z) \\
\widehat{S}_{13} = (F_x u_y + F_y u_x) & \widehat{S}_{14} = (F_y u_z + F_z u_y) & \widehat{S}_{15} = (F_x u_z + F_z u_x) \\
\widehat{S}_{16} = 0 & \widehat{S}_{17} = 0 & \widehat{S}_{18} = 0
\end{array}$$

The macroscopic variables, such as density ρ, velocity $\boldsymbol{u}$, and pressure p^*, can be expressed as:

$$\rho = \sum_\alpha f_\alpha \tag{H–3}$$

$$\rho \boldsymbol{u} = \sum_\alpha \boldsymbol{e}_\alpha f_\alpha + \frac{1}{2}\frac{\boldsymbol{F}_{ext}\delta t}{\gamma} \tag{H–4}$$

$$p^* = \gamma c_s^2 \rho$$

and the modified transport coefficients as:

$$\nu = \frac{\gamma}{3}\left[\frac{1}{s_k} - \frac{1}{2}\right]\delta t, \qquad k = 9, 11, 13, 14, 15$$

$$\zeta = \frac{2\gamma}{9}\left[\frac{1}{s_1} - \frac{1}{2}\right]\delta t$$

For SRT–LBM, the preconditioning leads to

$$f_\alpha(\boldsymbol{x} + \boldsymbol{e}_\alpha \delta t) - f_\alpha(\boldsymbol{x}, t) = -\frac{1}{\tau'}[f_\alpha(\boldsymbol{x}, t) - f_\alpha^{eq}(\boldsymbol{x}, t)] + (1 - \frac{1}{2\tau'})S'_\alpha \delta t \tag{H–5}$$

$$f_\alpha^{eq} = w_\alpha \rho\left[1 + \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4\gamma} - \frac{u^2}{2c_s^2\gamma}\right], \qquad c_s^2 = \frac{1}{3}c^2 \tag{H–6}$$

where the source term is given by

$$S'_\alpha = w_\alpha\left[\frac{\left(e_{\alpha k} - \frac{u_k}{\gamma}\right)}{c_s^2} + \frac{e_{\alpha\ell}u_\ell}{c_s^4\gamma}e_{\alpha k}\right]\frac{F_{ext,k}}{\gamma} \tag{H–7}$$

Extension to MHD flows

To accelerate steady-state convergence for simulations of flows with MHD effects, the magnetic induction equation is modified to become:

$$\frac{\partial \boldsymbol{B}}{\partial t} + \frac{\chi}{\gamma_m}\boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{u}) = \frac{1}{\gamma_m}\boldsymbol{\nabla} \cdot (\eta \boldsymbol{\nabla}\boldsymbol{B}) \tag{H–8}$$

In this equation, γ_m is a preconditioning parameter for the magnetic induction equation. The parameter χ is a scaling factor and is varied to adjust the effective Prandtl number, $\mathrm{Pr}_{\mathrm{m,eff}} = \chi \mathrm{Pr}_{\mathrm{m}}$.

The equation for the distribution function h is

$$h_{\alpha i}(\boldsymbol{x} + \boldsymbol{e}_\alpha^m \delta t, t + \delta t) - h_{\alpha i}(\boldsymbol{x}, t) - \frac{1}{\tau_m}\left(h_{\alpha i}(\boldsymbol{x}, t) - h_{\alpha i}^{eq}(\boldsymbol{x}, t)\right) \tag{H–9}$$

where $h^{eq}_{\alpha i}$ is modified due to preconditioning and Prandtl number scaling to become

$$h^{eq}_{\alpha i} = w_\alpha \left[B_i + \frac{4e^m_{\alpha j}}{c^2} \frac{\chi}{\gamma_m} \left(u_j B_i - u_i B_j \right) \right] \qquad \text{(H–10)}$$

The magnetic resistivity, η_m, is related to the modified relaxation time τ'_m through

$$\eta_m = \gamma_m \frac{c^2}{4} \left(\tau'_m - \frac{1}{2} \right) \delta t$$

The current density $\boldsymbol{J}$ can be calculated by taking into account the magnetic Prandtl number scaling:

$$\begin{aligned} \boldsymbol{J}_k &= \frac{1}{\chi} \frac{1}{\mu_m} \left(\boldsymbol{\nabla} \times \boldsymbol{B} \right)_k \\ &= -\frac{1}{\chi} \frac{4}{\mu_m c^2 \tau'_m} \epsilon_{ijk} \sum_{\alpha=0}^{b_m} \left(e^m_{\alpha i} h_{\alpha j} - e^m_{\alpha i} h^{eq}_{\alpha j} \right) \end{aligned} \qquad \text{(H–11)}$$

where

$$\sum_{\alpha=0}^{b_m} e^m_{\alpha j} h^{eq}_{\alpha i} = \frac{\chi}{\gamma_m} \left(u_j B_i - B_j u_i \right) \qquad \text{(H–12)}$$

and the Lorentz force that appears in the preconditioned MRT-LBM for flow computation is:

$$F_{\mathrm{Lorentz}} = \boldsymbol{J} \times \boldsymbol{B} \qquad \text{(H–13)}$$

More details of the preconditioned MRT–LBM formulation and results, including MHD effects, can be found in Premnath and Pattison (2007).

I Use of Nonuniform Grids in MRT–LBM through Second-Order Lagrangian Upwind Interpolation

In the following, we discuss an example of the post- interpolated-streaming step value of the distribution function corresponding to the direction $\alpha = 7$ for the D3Q19 lattice shown in Fig. 2. Considering second-order upwind interpolations along the respective Cartesian coordinate directions, we get (see Fig.37)

$$\begin{aligned} f_{7,i,j,k} &= a^-_{1,i,j,k} \left[b^-_{1,i,j,k} \tilde{f}_{7,i,j,k} + b^-_{2,i,j-1,k} \tilde{f}_{7,i,j-1,k} + b^-_{3,i,j-2,k} \tilde{f}_{7,i,j-2,k} \right] \\ &+ a^-_{2,i,j,k} \left[b^-_{1,i-1,j,k} \tilde{f}_{7,i-1,j,k} + b^-_{2,i-1,j-1,k} \tilde{f}_{7,i-1,j-1,k} + b^-_{3,i-1,j-2,k} \tilde{f}_{7,i-1,j-2,k} \right] \\ &+ a^-_{3,i,j,k} \left[b^-_{1,i-2,j,k} \tilde{f}_{7,i-2,j,k} + b^-_{2,i-2,j-1,k} \tilde{f}_{7,i-2,j-1,k} + b^-_{3,i-2,j-2,k} \tilde{f}_{7,i-2,j-2,k} \right] \end{aligned} \qquad \text{(I–1)}$$

Here, the “tilde” refers to the post-“effective” collision values of the distribution function (i.e., including forcing terms) obtained by executing the collision step in moment space in the MRT–LBM as discussed in Appendix F. The upwind Lagrange interpolation coefficients in Eq. (I–1) are given below:

$$a^-_{1,i,j,k} = \left[\frac{X_{m(i,j,k)} - \left(X_{m(i-1,j,k)} + \delta x \right)}{\left(X_{m(i,j,k)} + \delta x \right) - \left(X_{m(i-1,j,k)} + \delta x \right)} \right] \left[\frac{X_{m(i,j,k)} - \left(X_{m(i-2,j,k)} + \delta x \right)}{\left(X_{m(i,j,k)} + \delta x \right) - \left(X_{m(i-2,j,k)} + \delta x \right)} \right] \qquad \text{(I–2)}$$

$$a^-_{2,i,j,k} = \left[\frac{X_{m(i,j,k)} - \left(X_{m(i,j,k)} + \delta x \right)}{\left(X_{m(i-1,j,k)} + \delta x \right) - \left(X_{m(i,j,k)} + \delta x \right)} \right] \left[\frac{X_{m(i,j,k)} - \left(X_{m(i-2,j,k)} + \delta x \right)}{\left(X_{m(i-1,j,k)} + \delta x \right) - \left(X_{m(i-2,j,k)} + \delta x \right)} \right] \qquad \text{(I–3)}$$

$$a^-_{3,i,j,k} = \left[\frac{X_{m(i,j,k)} - \left(X_{m(i,j,k)} + \delta x \right)}{\left(X_{m(i-2,j,k)} + \delta x \right) - \left(X_{m(i,j,k)} + \delta x \right)} \right] \left[\frac{X_{m(i,j,k)} - \left(X_{m(i-1,j,k)} + \delta x \right)}{\left(X_{m(i-2,j,k)} + \delta x \right) - \left(X_{m(i-1,j,k)} + \delta x \right)} \right] \qquad \text{(I–4)}$$

and

$$b^{-}_{1,i,j,k} = \left[\frac{Y_{m(i,j,k)} - \left(Y_{m(i,j-1,k)} + \delta y\right)}{\left(Y_{m(i,j,k)} + \delta y\right) - \left(Y_{m(i,j-1,k)} + \delta y\right)}\right]\left[\frac{Y_{m(i,j,k)} - \left(Y_{m(i,j-2,k)} + \delta y\right)}{\left(Y_{m(i,j,k)} + \delta y\right) - \left(Y_{m(i,j-2,k)} + \delta y\right)}\right] \tag{I–5}$$

$$b^{-}_{2,i,j,k} = \left[\frac{Y_{m(i,j,k)} - \left(Y_{m(i,j,k)} + \delta y\right)}{\left(Y_{m(i,j-1,k)} + \delta y\right) - \left(Y_{m(i,j,k)} + \delta y\right)}\right]\left[\frac{Y_{m(i,j,k)} - \left(Y_{m(i,j-2,k)} + \delta y\right)}{\left(Y_{m(i,j-1,k)} + \delta y\right) - \left(Y_{m(i,j-2,k)} + \delta y\right)}\right] \tag{I–6}$$

$$b^{-}_{3,i,j,k} = \left[\frac{Y_{m(i,j,k)} - \left(Y_{m(i,j,k)} + \delta y\right)}{\left(Y_{m(i,j-2,k)} + \delta y\right) - \left(Y_{m(i,j,k)} + \delta y\right)}\right]\left[\frac{Y_{m(i,j,k)} - \left(Y_{m(i,j-1,k)} + \delta y\right)}{\left(Y_{m(i,j-2,k)} + \delta y\right) - \left(Y_{m(i,j-1,k)} + \delta y\right)}\right] \tag{I–7}$$

Similarly coefficients $c^{-}_{1,i,j,k}$, $c^{-}_{2,i,j,k}$, $c^{-}_{3,i,j,k}$, can be developed for the z direction which are needed for certain other particle directions. In the above, δx, δy and δz, are the streaming distances based on the particle velocity c during a time interval δt, i.e. $\delta x = c\delta t$, $\delta y = c\delta t$, $\delta z = c\delta t$. $X_{m(i,j,k)}$, $Y_{m(i,j,k)}$ and $Z_{m(i,j,k)}$ are the physical coordinates of grids. In general, the grids would be stretched such that $X_{m(i,j,k)} - X_{m(i-1,j,k)} \neq \delta x$, $Y_{m(i,j,k)} - Y_{m(i-1,j,k)} \neq \delta y$ and $Z_{m(i,j,k)} - Z_{m(i-1,j,k)} \neq \delta z$. In a similar way, if the particle velocity directions are flipped as compared to the above, as in for e.g., in the calculation of the interpolated-streaming values of $f_{8,i,j,k}$ (see Fig. 2), the respective interpolation coefficients are then based on the corresponding upwind directions, i.e. $a^{+}_{1,i,j,k}$, $a^{+}_{2,i,j,k}$ and $a^{+}_{3,i,j,k}$, and $b^{+}_{1,i,j,k}$, $b^{+}_{2,i,j,k}$ and $b^{+}_{3,i,j,k}$.

J Initial Perturbation of Velocity for Turbulent Channel Flow Simulations

The initial velocity perturbation employed for turbulent channel flow simulations is (Lam, 1990)

$$u_0^{'} = \epsilon L_1 \sin\left(\frac{2\pi z}{L_3}\right)\left[\cos\left(\frac{2\pi x}{L_1}\right)\sin\left(\frac{2\pi y}{L_2}\right) + \frac{1}{2}\cos\left(\frac{4\pi x}{L_1}\right)\sin\left(\frac{2\pi y}{L_2}\right) + \cos\left(\frac{2\pi x}{L_1}\right)\sin\left(\frac{4\pi y}{L_2}\right)\right]$$

$$v_0^{'} = -\epsilon L_2 \sin\left(\frac{2\pi z}{L_3}\right)\left[\frac{1}{2}\sin\left(\frac{2\pi x}{L_1}\right)\cos\left(\frac{2\pi y}{L_2}\right) + \frac{1}{2}\sin\left(\frac{4\pi x}{L_1}\right)\cos\left(\frac{2\pi y}{L_2}\right) + \frac{1}{4}\sin\left(\frac{2\pi x}{L_1}\right)\cos\left(\frac{4\pi y}{L_2}\right)\right]$$

$$w_0^{'} = -\epsilon L_3 \left[1 + \cos\left(\frac{2\pi z}{L_3}\right)\right]\left[\sin\left(\frac{2\pi x}{L_1}\right)\sin\left(\frac{2\pi y}{L_2}\right) + \sin\left(\frac{4\pi x}{L_1}\right)\sin\left(\frac{2\pi y}{L_2}\right) + \sin\left(\frac{2\pi x}{L_1}\right)\sin\left(\frac{4\pi y}{L_2}\right)\right]$$

where L_1, L_2 and L_3 are the streamwise, spanwise and wall-normal lengths of the computational domain.

These initial perturbations contain products of the first odd and even modes and satisfy continuity. In general, the solutions (in the statistical sense) should not depend on the particular choice of initial conditions. However, it has been found that if the above initial perturbations are used without the second or third terms, not all higher Fourier modes will be excited through nonlinear interactions in subsequent time steps.

K Derivation of Generalised Magnetic Induction Equation with Spatially Varying Electrical Conductivities

Here we provide the derivation of the magnetic induction equation with variable electrical conductivities (Premnath, 2006). Consider Faradays law relating the dynamics of magnetic induction $\boldsymbol{B}$ to the electric field $\boldsymbol{E}$

$$\partial_t \boldsymbol{B} = -\boldsymbol{\nabla} \times \boldsymbol{E} \tag{K–1}$$

When Ohm's law

$$\boldsymbol{E} = \frac{\boldsymbol{J}}{\sigma} - \boldsymbol{u} \times \boldsymbol{B} \tag{K–2}$$

is applied to Eq. (K–1), where $\boldsymbol{J}$ is the current density, $\sigma = \sigma(\vec{x})$ is the spatially varying electrical conductivity and $\boldsymbol{u}$ is the velocity field, it becomes:

$$\partial_t \boldsymbol{B} = -\boldsymbol{\nabla} \times \left(\frac{\boldsymbol{J}}{\sigma} - \boldsymbol{u} \times \boldsymbol{B} \right) \tag{K–3}$$

or

$$\partial_t \boldsymbol{B} - \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) = \boldsymbol{\nabla} \times \left(\frac{\boldsymbol{J}}{\sigma} \right) \tag{K–4}$$

Now, let us apply the Ampère-Maxwell law

$$\boldsymbol{J} = \frac{1}{\mu_m} \boldsymbol{\nabla} \times \boldsymbol{B} \tag{K–5}$$

where μ_m is the magnetic permeability. Let us first simplify the left hand side of Eq. (K–4). Using the vector identity:

$$\boldsymbol{\nabla} \times (\boldsymbol{A} \times \boldsymbol{B}) = \boldsymbol{A}(\boldsymbol{\nabla} \cdot \boldsymbol{B}) - \boldsymbol{B}(\boldsymbol{\nabla} \cdot \boldsymbol{A}) + (\boldsymbol{B} \cdot \boldsymbol{\nabla})\boldsymbol{A} - (\boldsymbol{A} \cdot \boldsymbol{\nabla})\boldsymbol{B} \tag{K–6}$$

and using divergence-free/solenoidal constraints on the velocity and magnetic fields,

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0; \qquad \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$$

$$-\boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) = -\left[(\boldsymbol{B} \cdot \boldsymbol{\nabla})\boldsymbol{u} - (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{B}\right] = -\left[\boldsymbol{\nabla} \cdot (\boldsymbol{B}\boldsymbol{u}) - \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{B})\right] = \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{u}) \tag{K–7}$$

Substituting Eq. (K-7) into Eq. (K-4), we get

$$\partial_t \boldsymbol{B} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{u}) = -\boldsymbol{\nabla} \times \left(\frac{1}{\sigma \mu_m} \boldsymbol{\nabla} \times \boldsymbol{B} \right) \tag{K–8}$$

let $\eta = 1/(\sigma \mu_m)$, where $\eta \equiv \eta(\vec{x})$ is usually called the magnetic resistivity, and using this, equation (K-8) may be rewritten as:

$$\partial_t \boldsymbol{B} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{u}) = -\boldsymbol{\nabla} \times (\eta \boldsymbol{\nabla} \times \boldsymbol{B}) \tag{K–9}$$

The right hand side of Eq. (K-9) can be simplified, while considering that the resistivity, η, is a spatially varying quantity. This is the magnetic induction equation with variable conductivity.

To simplify Eq. (K-9), let us first rewrite it in index notation. Then we will explicitly provide the components in compact form, from which the components in Cartesian coordinates may be readily obtained. Considering the usual summation convention for the repeated (i.e. dummy) indices, and noting that cross-products can be written in terms of permutation tensor, we write Eq. (K-9) in index notation:

$$\partial_t B_i + \partial_j (u_j B_i - B_j u_i) = S_i, \tag{K–10}$$

where

$$S_i = -\epsilon_{ijk} \partial_j \{ \eta \epsilon_{klm} \partial_l B_m \}. \tag{K–11}$$

Here, ϵ_{ijk} is the Levi-Civita permutation tensor, whose value for even and odd permutations, when indices i, j and k are distinct, are +1 and -1 respectively; and otherwise zero. Indices i through m represent one of the Cartesian component directions, i.e. $\{i, j, ..., m\} \in \{x, y, z\}$. Rewriting equation (K–11) and considering the permuting property of the permutation tensor, we get:

$$S_i = -\epsilon_{ijk} \epsilon_{klm} \partial_j \{ \eta \partial B_m \} = \epsilon_{kij} \epsilon_{klm} \partial_j \{ \eta \partial B_m \} \tag{K–12}$$

Now, we employ the well known identity of the permutation tensor products

$$\epsilon_{kpq}\epsilon_{krs} = \delta_{pr}\delta_{qs} - \delta_{ps}\delta_{qr} \tag{K–13}$$

in Eq. (K–12). The terms on the right hand side of Eq. (K–13) such as δ_{pr} are the Kronecker delta tensor, sometimes also called the substitution tensor, whose value is unity if both the indices are equal and zero otherwise. We get

$$S_i = -\epsilon_{kij}\epsilon_{klm}\partial_j\{\eta\partial B_m\} = -(\delta_{il}\delta_{jm} - \delta_{im}\delta_{jl})\partial_j\{\eta\partial B_m\} = -\delta_{il}\delta_{jm}\partial_j\{\eta\partial B_m\} + \delta_{im}\delta_{jl}\partial_j\{\eta\partial B_m\} \tag{K–14}$$

Using the substitution property of the Kronecker delta tensor, i.e. $\delta_{il}\Gamma_l = \Gamma_i$, in equation Eq. (K–14), we can considerably simplify it to:

$$S_i = -\partial_j\{\eta\partial_i B_j\} + \partial_j\{\eta\partial_j B_i\} = \partial_j\{\eta\partial_j B_i\} - \partial_j\{\eta\partial_i B_j\} \tag{K–15}$$

When the divergence-free condition for the magnetic induction is imposed, i.e. $\partial_j B_j = 0$, the term $\partial_j\{\eta\partial_i B_j\}$ in Eq. (K–15) modifies to $(\partial_i B_j)(\partial_j\eta)$. So, an alternative form of Eq. (K–15) is

$$S_i = \partial_j\{\eta\partial_j B_i\} - (\partial_i B_j)(\partial_j\eta) \tag{K–16}$$

Thus, substituting Eqs. (K–15) and (K–16) into Eq. (K–10), we obtain two equivalent forms for the *generalised* induction equation in index notation

$$\partial_t B_i + \partial_j(u_j B_i - B_j u_i) = \partial_j(\eta\partial_j B_i) - \partial_j(\eta\partial_i B_j) \tag{K–17}$$

$$\partial_t B_i + \partial_j(u_j B_i - B_j u_i) = \partial_j(\eta\partial_j B_i) - (\partial_i B_j)(\partial_j\eta) \tag{K–18}$$

The final term in Eqs. (K–17) and (K–18) is the additional term with respect to the standard induction equation for constant conductivity, i.e.

$$\partial_t B_i + \partial_j(u_j B_i - B_j u_i) = \partial_j(\eta\partial_j B_i) \tag{K–19}$$

Thus, to properly account for spatial variation of electrical conductivities or magnetic diffusivities, it appears from this short note that additional terms that are included on the right of Eqs. (K–17) and (K–18) should be added to the standard form of the magnetic induction equation, Eq. (29).

To clarify things further, we provide in the following the expanded forms of Eqs. (K–17) and (K–18) for three-dimensional (3D) problems. The Cartesian components corresponding to Eq. (K–17) are:

$$\begin{aligned}\partial_t B_x &+ \partial_y(u_y B_x - B_y u_x) + \partial_z(u_z B_x - B_z u_x) = \partial_x(\eta\partial_x B_x) + \partial_y(\eta\partial_y B_x) + \partial_z(\eta\partial_z B_x)\\ &- [\partial_x(\eta\partial_x B_x) + \partial_y(\eta\partial_x B_y) + \partial_z(\eta\partial_x B_z)]\end{aligned} \tag{K-20a}$$

$$\begin{aligned}\partial_t B_y &+ \partial_x(u_x B_y - B_x u_y) + \partial_z(u_z B_y - B_z u_y) = \partial_x(\eta\partial_x B_y) + \partial_y(\eta\partial_y B_y) + \partial_z(\eta\partial_z B_y)\\ &- [\partial_x(\eta\partial_y B_x) + \partial_y(\eta\partial_y B_y) + \partial_z(\eta\partial_y B_z)]\end{aligned} \tag{K-20b}$$

$$\begin{aligned}\partial_t B_z &+ \partial_x(u_x B_z - B_x u_z) + \partial_y(u_y B_z - B_y u_z) = \partial_x(\eta\partial_x B_z) + \partial_y(\eta\partial_y B_z) + \partial_z(\eta\partial_z B_z)\\ &- [\partial_x(\eta\partial_z B_x) + \partial_y(\eta\partial_z B_y) + \partial_z(\eta\partial_z B_z)]\end{aligned} \tag{K-20c}$$

Similarly, we write the components corresponding to the equivalent generalized induction equation, i.e. Eq. (K–18) as

$$\begin{aligned}\partial_t B_x &+ \partial_y(u_y B_x - B_y u_x) + \partial_z(u_z B_x - B_z u_x)\\ &= \partial_x(\eta\partial_x B_x) + \partial_y(\eta\partial_y B_x) + \partial_z(\eta\partial_z B_x)\\ &- [(\partial_x B_x)(\partial_x\eta) + (\partial_x B_y)(\partial_y\eta)\partial_x B_y) + (\partial_x B_z)(\partial_z\eta)]\end{aligned} \tag{K-21a}$$

$$\begin{aligned} \partial_t B_y &+ \partial_x(u_x B_y - B_x u_y) + \partial_z(u_z B_y - B_z u_y) \\ &= \partial_x(\eta \partial_x B_y) + \partial_y(\eta \partial_y B_y) + \partial_z(\eta \partial_z B_y) \\ &- [(\partial_y B_x)(\partial_x \eta) + (\partial_y B_y)(\partial_y \eta)\partial_x B_y) + (\partial_y B_z)(\partial_z \eta)] \end{aligned} \tag{K-21b}$$

$$\begin{aligned} \partial_t B_z &+ \partial_x(u_x B_z - B_x u_z) + \partial_y(u_y B_z - B_y u_z) \\ &= \partial_x(\eta \partial_x B_z) + \partial_y(\eta \partial_y B_z) + \partial_z(\eta \partial_z B_z) \\ &- [(\partial_z B_x)(\partial_x \eta) + (\partial_z B_y)(\partial_y \eta)\partial_x B_y) + (\partial_z B_z)(\partial_z \eta)] \end{aligned} \tag{K-21c}$$

As before, the terms within the square brackets in Eqs. (K-20) and (K-21) represent the additional terms with respect to the standard induction equation for constant conductivity. Inspection of these two equations indicates that Eq. (K-20) is in conservative-like form and may be more suitable for numerical implementation for multi-material magnetohydrodynamic (MHD) problems that involve use of variable conductivities.

If L and U_0 are the length and velocity scales, respectively, then $Re_m = U_0 L/\eta = \mu_m \sigma U_0 L$ is the magnetic Reynolds number. For liquid metal MHD, generally, $Re_m \ll 1$ and so the total magnetic induction B_i may be split into the applied field, B_i^A and induced field B_i^I, i.e. $B_i = B_i^A + B_i^I$. In such cases, the generalized induction equations for the total field, i.e. Eqs. (K-20) or (K-21), may be reduced in terms of induced field as the unknown quantity by neglecting higher order terms of the smaller quantities, i.e. B_i^I so as to simplify their numerical solution. We thus obtain the "reduced form" of the generalized induction equation from Eqs. (K-20) and (K-21), respectively, as

$$\begin{aligned} &\partial_t B_x^I + \partial_y(u_y B_x^A - B_y^A u_x) + \partial_z(u_z B_x^A - B_z^A u_x) \\ &\quad = \partial_x(\eta \partial_x B_x^I) + \partial_y(\eta \partial_y B_x^I) + \partial_z(\eta \partial_z B_x^I) - [\partial_x(\eta \partial_x B_x^I) + \partial_y(\eta \partial_x B_y^I) + \partial_z(\eta \partial_x B_z^I)] \end{aligned} \tag{K-22a}$$

$$\begin{aligned} &\partial_t B_y^I + \partial_x(u_x B_y^A - B_x^A u_y) + \partial_z(u_z B_y^A - B_z^A u_y) \\ &\quad = \partial_x(\eta \partial_x B_y^I) + \partial_y(\eta \partial_y B_y^I) + \partial_z(\eta \partial_z B_y^I) - [\partial_x(\eta \partial_y B_x^I) + \partial_y(\eta \partial_y B_y^I) + \partial_z(\eta \partial_y B_z^I)] \end{aligned} \tag{K-22b}$$

$$\begin{aligned} &\partial_t B_z^I + \partial_x(u_x B_z^A - B_x^A u_z) + \partial_y(u_y B_z^A - B_y^A u_z) \\ &\quad = \partial_x(\eta \partial_x B_z^I) + \partial_y(\eta \partial_y B_z^I) + \partial_z(\eta \partial_z B_z^I) - [\partial_x(\eta \partial_z B_x^I) + \partial_y(\eta \partial_z B_y^I) + \partial_z(\eta \partial_z B_z^I)] \end{aligned} \tag{K-22c}$$

and

$$\begin{aligned} \partial_t B_x^I + \partial_y(u_y B_x^A - B_y^A u_x) &+ \partial_z(u_z B_x^A - B_z^A u_x) \\ &= \partial_x(\eta \partial_x B_x^I) + \partial_y(\eta \partial_y B_x^I) + \partial_z(\eta \partial_z B_x^I) \\ &- [(\partial_x B_x^I)(\partial_x \eta) + (\partial_x B_y^I)(\partial_y \eta)\partial_x B_y^I) + (\partial_x B_z^I)(\partial_z \eta)] \end{aligned} \tag{K-23a}$$

$$\begin{aligned} \partial_t B_y^I + \partial_x(u_x B_y^A - B_x^A u_y) &+ \partial_z(u_z B_y^A - B_z^A u_y) \\ &= \partial_x(\eta \partial_x B_y^I) + \partial_y(\eta \partial_y B_y^I) + \partial_z(\eta \partial_z B_y^I) \\ &- [(\partial_y B_x^I)(\partial_x \eta) + (\partial_y B_y^I)(\partial_y \eta)\partial_x B_y^I) + (\partial_y B_z^I)(\partial_z \eta)] \end{aligned} \tag{K-23b}$$

$$\begin{aligned} \partial_t B_z^I + \partial_x(u_x B_z^A - B_x^A u_z) &+ \partial_y(u_y B_z^A - B_y^A u_z) \\ &= \partial_x(\eta \partial_x B_z^I) + \partial_y(\eta \partial_y B_z^I) + \partial_z(\eta \partial_z B_z^I) \\ &- [(\partial_z B_x^I)(\partial_x \eta) + (\partial_z B_y^I)(\partial_y \eta)\partial_x B_y^I) + (\partial_z B_z^I)(\partial_z \eta)] \end{aligned} \tag{K-23c}$$

As before, between these two equations, Eq. (K-22) is more likely to be suitable for numerical solution owing to its "conservative-like" form.

It may be noted that while the additional terms with respect to the standard form may have influence for the general case of 3D problems, a simple analysis would show that they dropout at least for the classical 1D Hartmann flows or for fully-developed MHD flows even when we consider variable conductivity in the direction transverse to the flow. In any case, solving the generalized set of equations for magnetic induction - full or reduced, which maintains all the electrodynamical laws for multi-material MHD problems intact and required in general situations, appears to be a very challenging task. Some important factors to be considered that are specific to numerical solution of generalized induction equation include: (i) the equations should be made available in conservative form, in which case the starting point could involve Eq. (K-20) or (K-21). (ii) In contrast to that of standard induction equation for constant conductivity, the components equations for the generalized case are fully "coupled" with one another; For e.g. inspection of Eq. (K-20a), i.e. the equation for component B_x shows that its RHS, through the additional terms within square brackets, contains all the possible components, i.e. B_x, B_y and B_z, which make the system of induction equations coupled, a factor of great importance when they are attempted to be solved using a fully-implicit procedure. Consideration of just these two factors implies that developing suitable numerical solution techniques for general cases would be quite challenging.

As an example, in the following we just make an attempt to obtain the finite difference discretization of the first additional term within square brackets in Eq. (K-20b). It is the cross-derivatives $[\partial_x(\eta\partial_y B_x)]$ in "quasi-conservative" form. In this first attempt, we use a naive strategy, i.e. full stencil instead of the usual half-stencil used for the inner derivative:

$$
\begin{aligned}
[\partial_x(\eta\partial_y B_x)]_{(i,j,k)} &\approx \frac{(\eta\partial_y B_x)_{(i+1,j,k)} - (\eta\partial_y B_x)_{(i-1,j,k)}}{x_{(i+1,j,k)} - x_{(i-1,j,k)}} \\
&= \frac{\eta_{(i+1,j,k)}(\partial_y B_x)_{(i+1,j,k)} - \eta_{(i-1,j,k)}(\partial_y B_x)_{(i-1,j,k)}}{x_{(i+1,j,k)} - x_{(i-1,j,k)}} \\
&= \frac{\eta_{(i+1,j,k)}\left[\frac{B_{x(i+1,j+1,k)}-B_{x(i+1,j-1,k)}}{y_{(i+1,j+1,k)}-y_{(i+1,j-1,k)}}\right]\eta_{(i-1,j,k)}\left[\frac{B_{x(i-1,j+1,k)}-B_{x(i-1,j-1,k)}}{y_{(i-1,j+1,k)}-y_{(i-1,j-1,k)}}\right]}{x_{(i+1,j,k)} - x_{(i-1,j,k)}}
\end{aligned} \qquad \text{(K-24)}
$$

This discretisation leads to variables residing only on diagonal nodes on the computational molecule, and does not use those that are on the primary coordinate directions. At the very least, in conjunction with the other discretised terms, this strategy entails solving on a larger computational stencil than that for the standard induction equation. It may also cause undesirable numerical consequences, such as chequer-boarding, as the variables on the principal coordinate directions are absent. An alternative attempt could be to try discretising the inner derivatives using a half stencil, i.e. at $(i + 1/2, j, k)$ and $(i - 1/2, j, k)$, and then use some form of averaging, such as tht used in finite-volume methods, to obtain the values at "half" nodes, e.g. $(i + 1/2, j + 1/2, k)$. In this latter strategy, we can introduce influence of variables at nodes along the principal coordinate directions. In any case, because of the coupled nature of the system of equations, solving their discretised versions using a fully implicit strategy could require use of "time-splitting" procedures. Other obvious, albeit simpler, possibilities include trying semi-implicit or explicit techniques, which could pose significant limitation on the Hartmann number of the flow that could be resolved. These difficulties mean unique opportunities for the development of suitable numerical techniques for the solution of the generalised induction equation – challenging tasks for the future.