

# Progress in lattice Boltzmann methods for magnetohydrodynamic flows relevant to fusion applications

M.J. Pattison<sup>a,\*</sup>, K.N. Premnath<sup>a,b</sup>, N.B. Morley<sup>c</sup>, M.A. Abdou<sup>c</sup>

<sup>a</sup> *MetaHeuristics LLC, 3944 State St., Ste. 350, Santa Barbara, CA 93105, USA*

<sup>b</sup> *UCSB, Chemical Engineering Department, Santa Barbara, CA 93106, USA*

<sup>c</sup> *UCLA, MAE Department, 44-114 Engineering IV, 420 Westwood Pza, Los Angeles, CA 90095-1597, USA*

Received 15 May 2007; received in revised form 27 September 2007; accepted 13 October 2007

Available online 26 December 2007

## Abstract

In this paper, an approach to simulating magnetohydrodynamic (MHD) flows based on the lattice Boltzmann method (LBM) is presented. The dynamics of the flow are simulated using a so-called multiple relaxation time (MRT) lattice Boltzmann equation (LBE), in which a source term is included for the Lorentz force. The evolution of the magnetic induction is represented by introducing a vector distribution function and then solving an appropriate lattice kinetic equation for this function. The solution of both distribution functions are obtained through a simple, explicit, and computationally efficient stream-and-collide procedure. The use of the MRT collision term enhances the numerical stability over that of a single relaxation time approach. To apply the methodology to solving practical problems, a new extrapolation-based method for imposing magnetic boundary conditions is introduced and a technique for simulating steady-state flows with low magnetic Prandtl number is developed. In order to resolve thin layers near the walls arising in the presence of high magnetic fields, a non-uniform gridding strategy is introduced through an interpolated-streaming step applied to both distribution functions. These advances are particularly important for applications in fusion engineering where liquid metal flows with low magnetic Prandtl numbers and high Hartmann numbers are introduced. A number of MHD benchmark problems, under various physical and geometrical conditions are presented, including 3-D MHD lid driven cavity flow, high Hartmann number flows and turbulent MHD flows, with good agreement with prior data. Due to the local nature of the method, the LBM also demonstrated excellent performance on parallel machines, with almost linear scaling up to 128 processors for a MHD flow problem.

© 2007 Elsevier B.V. All rights reserved.

PACS: 47.65.+a

Keywords: Magnetohydrodynamics; Lattice Boltzmann method; High Hartmann number flows; Parallel computing

## 1. Introduction

The flow of an electrically conducting fluid in a magnetic field is influenced by magnetohydrodynamic (MHD) forces resulting from the interaction of induced electric currents with the applied magnetic field. In nature, systems in which MHD effects are important include the Earth's core and solar flares, and in the engineering world, the electromagnetic casting of metals and the confinement of plasmas. Another area of much interest is fusion engineering; reactor designs commonly involve the use of electrically conducting liquid metals. For example, thermal blankets

usually use liquid metals to transport heat [1], and some reactor designs use flowing liquid metals on the first wall [2,3]. The high magnetic fields used to confine the plasma result in large MHD forces, whose effects dominate the flow in such components. The study of MHD has therefore been of particular importance to fusion engineering, forming significant components of facilities and projects such as APEX and ITER [4].

There are a number of different techniques for the numerical simulation of fluid flows, most of which are based on finite difference or finite volume methods, in which the continuum conservation equations are discretised and solved on computational grids. These methods can be extended to MHD flows by introducing a solution procedure for the magnetic induction equation and including a term for the Lorentz force in the equations for the flow dynamics [3]. An alternative approach,

\* Corresponding author.

E-mail address: [martin@metah.com](mailto:martin@metah.com) (M.J. Pattison).

which is becoming increasingly popular, is the lattice Boltzmann method (LBM) [5]. The LBM has its roots in kinetic theory, and the general idea behind this scheme is to compute a probability distribution function  $f_\alpha(\mathbf{x}, t)$ , where  $f_\alpha$  is the population of particles, representing fluid elements with a velocity along the direction  $\alpha$  at position  $\mathbf{x}$  and time  $t$  as they move and collide on a lattice. The collective behaviour of the distribution of particle populations represents that of the dynamics of fluid flow. A number of different schemes for calculating the evolution of the distribution function are available; most codes use single relaxation time (SRT) models for representing the effects of collisions [6–8]. However, the most advanced formulations use the recently-developed multiple relaxation time (MRT) model [9,10]. This latter method has better accuracy and numerical stability than the SRT models, and was chosen for the work described here.

The LBM can be extended to handle MHD flows, with the magnetic induction equation being solved in a manner similar to that for the fluid flow [12,13]. A vector distribution function representing the magnetic field is introduced, and a Boltzmann-type kinetic equation is constructed such that the behaviour of the particle populations corresponds to that of the induction equation. This can then be discretised and solved on a lattice grid.

One advantage of the lattice Boltzmann method is that it is very well suited to parallel processing on supercomputers with distributed memory architectures. At each time step, it is only necessary for a processor to communicate with a fixed number of other processors, typically between two and eight depending on how the computational domain is divided up, and for large problems, the execution speed is roughly proportional to the number of processors used. In fact, an LBM code has been successfully used to simulate certain unbounded MHD flows relevant to plasma physics on 4800 processors [14]. In contrast, with other methods each processor has to communicate with all, or many other, processors involved in the computation; in this situation, when large numbers of processors are used, the time spent communicating can severely reduce the execution speed compared with the LBM where the execution speed is almost proportional to the number of processors used.

A number of new developments for the application of the LBM to MHD problems are presented here. An extrapolation-based technique for scalar distribution functions used to apply velocity boundary conditions has been adapted to a vector distribution function to allow the magnetic field to be specified at boundaries. In contrast to earlier methods [12], the imposed magnetic field can be at any angle to the axes. A preconditioning method has been applied to the induction equation to enable the simulation of fluids with very low magnetic Prandtl numbers, such as liquid metals. This technique can also be used to accelerate solution convergence for steady state flows. A means of allowing non-uniform grids to be used with both vector and scalar distribution functions in 3D has been formulated to allow higher resolution of regions with high gradients, such as Hartmann layers. All these new developments enable the LBM to be used for problems in fusion engineering which commonly involve flows of liquid metals in high magnetic fields.

As well as discussing the models used for MHD computations, this paper presents the results of validations for a wide range of canonical flow problems where analytical solutions exist or results from other prior computational fluid dynamics (CFD) simulations are available. Following confirmation of the validity of the models used, tests which demonstrate the efficiency of the LBM on parallel computers with large numbers of processors are presented and the effectiveness of using stretched grids is investigated. Additional examples of the LBM results presented here are the large eddy simulation (LES) of turbulent MHD flow through a pipe, and a simulation of a section of a thermal blanket relevant to fusion applications.

## 2. Computational approach

### 2.1. Multiple relaxation time lattice Boltzmann equation for hydrodynamic fields

The lattice Boltzmann method is a relatively recent computational approach based on kinetic theory for solving fluid mechanics and other physical problems [5]. In brief, the LBM consists of solving the lattice Boltzmann equation for the evolution of a distribution function  $f(\mathbf{x}, \mathbf{v}, t)$  as they move and collide on a lattice. The solution of the equation involves two main steps, which represent streaming and collision of particle populations. Usually the collision process is represented by the Bhatnagar–Gross–Krook model [6] where the particle populations relax to a local equilibrium state at a rate determined by a characteristic relaxation time parameter. More recently, the multiple relaxation time model has been introduced, which involves several relaxation times and proves to have substantially better numerical stability [9].

The LBE is discretised and solved on a lattice grid. The number of discrete velocity directions representing the lattice is chosen to respect certain symmetry needed to recover the isotropy of the viscous stress tensor of the fluid flow [5]. In three dimensional implementations, a cubic grid is used and 15 or 19 particle velocity models are commonly used, though other models are occasionally found. In this work, the 19 velocity model was used, due to its superior numerical stability, and this is shown in Fig. 1.

The vectors corresponding to this three-dimensional, nineteen velocity (D3Q19) particle velocities set are given by:

$$e_\alpha = \begin{cases} (0, 0, 0), & \alpha = 1, \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1), & \alpha = 2, \dots, 7, \\ (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1), & \alpha = 8, \dots, 19. \end{cases} \quad (1)$$

The Cartesian component  $c$  of the particle velocity  $e_\alpha$  is given by  $c = \delta_x / \delta_t$ , where  $\delta_x$  is the lattice spacing and  $\delta_t$  the time step. In this paper, the convention that Greek symbols ( $\alpha, \beta, \dots$ ) will be used to represent particle velocities directions and Roman symbols ( $i, j, k$ ) for Cartesian directions will be used. The corresponding vector of distribution functions  $f$  at a location may be written as

$$f = [f_0, f_1, f_2, \dots, f_{18}]^T. \quad (2)$$

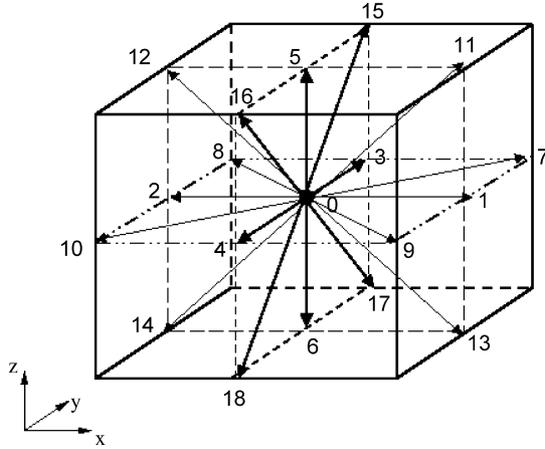


Fig. 1. Three dimensional, 19 velocity (D3Q19) lattice.

As previously mentioned, the distribution function is calculated using a two-step procedure comprising a so-called collision step and a streaming step. In the most advanced formulation currently available, the MRT-LBE with forcing terms are used [9,10]:

$$\begin{aligned} \tilde{f}_\alpha(\mathbf{x}, t) - f_\alpha(\mathbf{x}, t) &= \sum_\beta \Lambda_{\alpha\beta} (f_\beta - f_\beta^{eq}) + \sum_\beta \left( I_{\alpha\beta} - \frac{1}{2} \Lambda_{\alpha\beta} \right) S_\beta \delta_t, \end{aligned} \quad (3-a)$$

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta_t, t + \delta_t) = \tilde{f}_\alpha(\mathbf{x}, t). \quad (3-b)$$

The first term on the right hand side (RHS) of Eq. (3-a) represents the cumulative effect of particle collisions on the evolution of the distribution function  $f_\alpha$ , and can be thought of as representing the effects of viscosity, as well as other processes. Collision is considered as a relaxation process in which  $f_\beta$  relaxes to its local equilibrium value  $f_\beta^{eq}$  at a rate determined by the relaxation time matrix  $\Lambda_{\alpha\beta}$ . The MRT model 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  $\rho$  and momentum  $\rho \mathbf{u}$ , while the stress tensors relax during collisions at rates determined by fluid properties such as the shear and bulk viscosities. Thus certain relaxation times forming components of the collision matrix  $\Lambda_{\alpha\beta}$  in the MRT model are developed to reflect the underlying physics, while those which do not affect hydrodynamics are chosen to enhance the numerical stability of the approach. For more details, the reader is referred to Refs. [9,10].

The second term on the RHS of Eq. (3-a) introduces changes in the evolution of distribution function from driving forces  $\mathbf{F}$ , such as gravity ( $\rho \mathbf{g}$ ) through a source term  $S_\alpha$ . For MHD flows, one must also include the Lorentz force,  $\mathbf{J} \times \mathbf{B}$ , where  $\mathbf{J}$  is the current density and  $\mathbf{B}$  is the magnetic field strength, through  $S_\alpha$ . This source term may be written as [10]:

$$S_\alpha = \frac{(\mathbf{e}_\alpha \cdot \mathbf{u} - u_j) F_j}{\rho c_s^2} f_\alpha^{eq,M}(\rho, \mathbf{u}), \quad (4)$$

where  $f_\alpha^{eq,M}(\rho, \mathbf{u})$  is the local Maxwellian, which is dependent on the local density and velocity as

$$\begin{aligned} f_\alpha^{eq,M}(\rho, \mathbf{u}) &= \left( 1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2} \right), \\ \omega_\alpha &= \begin{cases} \frac{1}{3} & \alpha = 1 \\ \frac{1}{18} & \alpha = 2, \dots, 7 \\ \frac{1}{36} & \alpha = 8, \dots, 19. \end{cases} \end{aligned} \quad (5)$$

and  $c_s = (1/\sqrt{3})c$  is the speed of sound of the model. In the second term of Eq. (3-a),  $\mathbf{I}_{\alpha\beta}$  is the identity matrix.

At this point 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 that the implementation of Eq. (3-a) does not involve direct summation as shown, but uses a highly optimised procedure that involves transformations into moment space and exploits certain properties of  $\Lambda_{\alpha\beta}$  [9–11]. With these optimisations, it is found that despite its much greater complexity, the MRT only takes about 10–30% more CPU time than the BGK model [9–11]. Thus Eq. (3-a) provides the post-collision value of the distribution function given by  $\tilde{f}_\alpha$ .

Equation (3-b) is known as the advection, or streaming, step and deals with the change in the distribution function during a time interval  $\delta_t$ , as the particles propagate from location  $\mathbf{x}$  to their adjacent location  $\mathbf{x} + \mathbf{e}_\alpha \delta_t$ , with a velocity  $\mathbf{e}_\alpha$  along the characteristic direction  $\alpha$ .

The local macroscopic density and velocity fields are then given by

$$\rho = \sum_{\alpha=0}^{18} f_\alpha, \quad (6)$$

$$\rho \mathbf{u} = \sum_{\alpha=0}^{18} f_\alpha \mathbf{e}_\alpha + \frac{1}{2} \mathbf{F} \delta_t, \quad (7)$$

and the pressure field  $p$  may be written as

$$p = c_s^2 \rho. \quad (8)$$

The behaviour of the populations represented by the distribution function  $\mathbf{f}$  corresponds to that of fluid flow, and the incompressible Navier–Stokes equations can be recovered from the lattice Boltzmann equations for the case of low Mach number ( $v \ll c_s$ , where  $c_s$  is the speed of sound). The discussion of the MRT model presented here has been relatively brief, but more detailed descriptions can be found elsewhere [9–11].

Two important points to note are that there is no pressure Poisson equation to solve, and that the solution scheme is explicit, with information required from neighbouring nodes only. The solution of the Poisson equation is time consuming and typically takes 80–90% of the CPU time in traditional

CFD solvers [15]; its absence means that LBM codes are relatively fast on a per time step per grid point basis. The explicit nature of the computations means that codes based on the LBM can run very efficiently on parallel architectures, and this is one of the main motivations for its use. These two advantages also hold when lattice Boltzmann methods are used for the magnetic induction equation, which is discussed in Section 2.3.

### 2.2. Boundary conditions for hydrodynamic fields

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

$$f_{\bar{\alpha}}(\mathbf{x}, t + \delta_t) = \tilde{f}_{\alpha}(\mathbf{x}, t), \tag{9}$$

where  $\bar{\alpha}$  is the direction opposite to  $\alpha$ . 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 [16–19]. 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, the so-called interpolated bounce back scheme [18] was selected, due to its good numerical stability. 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. 2).

The first-order implementation of the scheme is:

$$f_{\bar{\alpha}}(\mathbf{x}, t + \delta_t) = 2q\tilde{f}_{\alpha}(\mathbf{x}, t) + (1 - 2q)\tilde{f}_{\bar{\alpha}}(\mathbf{x} - e_{\alpha}\delta_t, t) \tag{10}$$

for  $q < 0.5$

$$f_{\bar{\alpha}}(\mathbf{x}, t + \delta_t) = \frac{1}{2q}\tilde{f}_{\alpha}(\mathbf{x}, t) + \frac{(2q - 1)}{2q}\tilde{f}_{\bar{\alpha}}(\mathbf{x}, t) \tag{11}$$

for  $q \geq 0.5$

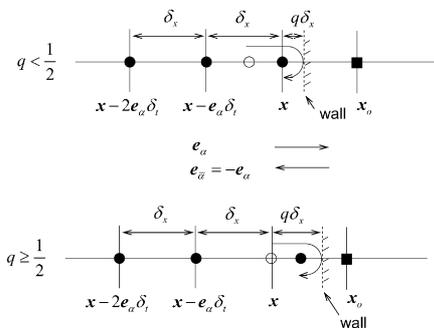


Fig. 2. Wall boundary implementation for cut-lattice cells.

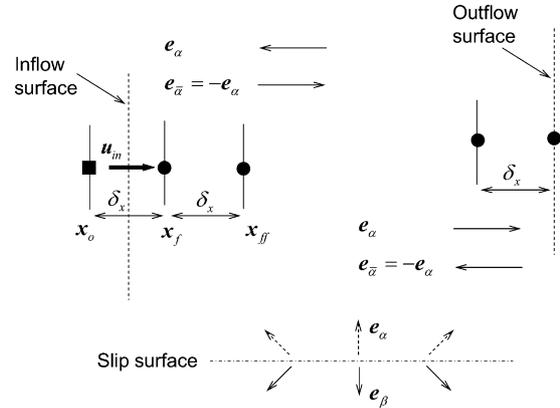


Fig. 3. Implementation of inflow, outflow, and symmetry boundary conditions.

and the second-order version is:

$$f_{\bar{\alpha}}(\mathbf{x}_f, t + \delta_t) = \frac{1}{q(2q + 1)}\tilde{f}_{\alpha}(\mathbf{x}_f, t) + (1 - 2q)(1 + 2q)\tilde{f}_{\bar{\alpha}}(\mathbf{x} - e_{\alpha}\delta_t, t) + q(1 - 2q)\tilde{f}_{\bar{\alpha}}(\mathbf{x} - 2e_{\alpha}\delta_t, t) \text{ for } q < 0.5 \tag{12}$$

$$f_{\bar{\alpha}}(\mathbf{x}_f, t + \delta_t) = \frac{1}{q(2q + 1)}\tilde{f}_{\alpha}(\mathbf{x}, t) + \frac{2q - 1}{q}\tilde{f}_{\bar{\alpha}}(\mathbf{x}, t) + \frac{1 - 2q}{1 + 2q}\tilde{f}_{\bar{\alpha}}(\mathbf{x} - e_{\alpha}\delta_t, t) \text{ for } q \geq 0.5 \tag{13}$$

In the present work, 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 scheme (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 [20] that adds appropriate momentum to the particle populations is implemented (see Fig. 3):

$$f_{\bar{\alpha}}(\mathbf{x}, t + \delta_t) = \tilde{f}_{\alpha}(\mathbf{x}, t) + 2\omega_{\alpha}\rho_{in}\frac{e_{\bar{\alpha}} \cdot u_{in}}{c_s^2} \tag{14}$$

For outflow boundaries, an extrapolation method is employed [21].

$$f_{\bar{\alpha}}(\mathbf{x}, t + \delta_t) = 2\tilde{f}_{\bar{\alpha}}(\mathbf{x}, t) - \tilde{f}_{\bar{\alpha}}(\mathbf{x} - e_{\alpha}\delta_t, t) \tag{15}$$

in which the equilibrium distributions 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  $f_{\beta}(\mathbf{x}, t + \delta_t) = \tilde{f}_{\bar{\alpha}}(\mathbf{x}, t)$ . After implementation of them, initial computations were carried out for a set of cases involving laminar flows without MHD effects to assess their suitability. These tests yielded results that were in excellent comparison with analytical solution and other computational results.

### 2.3. Vector lattice Boltzmann equation for magnetic induction fields

For flows with MHD effects, it is necessary to calculate the magnetic field, and this can be done through the magnetic induction equation [22,23]:

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{B} - \mathbf{B}\mathbf{u}) = \eta \nabla^2 \mathbf{B} \quad (16)$$

where  $\mathbf{B}$  is the magnetic induction field,  $\eta$  is the magnetic resistivity or diffusivity, defined by  $\eta = 1/(\sigma\mu)$ , and  $\sigma$  and  $\mu$  are the electrical conductivity and the magnetic permeability, respectively.

Within the LBM framework, the equation for the magnetic field can be solved in a similar fashion to the fluid dynamic equations, in which the evolution of a distribution function is computed. This approach has previously been taken by Del-lar [12], who successfully performed a set of two-dimensional MHD simulations, and forms the basis of the present work. To solve the magnetic induction equation with this approach, a vector distribution function  $\mathbf{h}$  is employed, since a scalar distribution function, as is used for the hydrodynamic equations, will not suffice [12]. A single relaxation time model is used for its evolution:

$$\tilde{\mathbf{h}}_\alpha(\mathbf{x}, t) = \mathbf{h}_\alpha(\mathbf{x}, t) - [\mathbf{h}_\alpha(\mathbf{x}, t) - \mathbf{h}_\alpha^{\text{eq}}(\mathbf{x}, t)]/\tau_m, \quad (17)$$

$$\mathbf{h}_\alpha(\mathbf{x} + \mathbf{e}_\alpha, \delta_t, t + \delta_t) = \tilde{\mathbf{h}}_\alpha(\mathbf{x}, t). \quad (18)$$

The values with a tilde represent post-collision values, as was the case with the hydrodynamic equations. The relaxation time,  $\tau_m$ , is a function of the conductivity and permeability of the medium and for the 3-D case they are related by [13]:

$$\tau_m = \frac{1}{2} + 4\eta \frac{\delta_t}{(\delta_x)^2}. \quad (19)$$

In practical implementations of this scheme, it is found that only seven directions are required and are represented by  $\mathbf{e}_\alpha^m$ ; 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 (Fig. 1).

The components of the equilibrium distribution functions  $\mathbf{h}_i^{\text{eq}}$  are then calculated through the following relation, given in index notation [13]:

$$h_{\alpha i}^{\text{eq}} = W_\alpha \left[ \hat{B}_i + \frac{4e_{\alpha j}^m}{c^2} (u_j \hat{B}_i - \hat{B}_j u_i) \right],$$

$$W_\alpha = \begin{cases} \frac{1}{4}, & \alpha = 0 \\ \frac{1}{8}, & \alpha = 1, \dots, 6 \end{cases} \quad (20)$$

where  $\hat{B}$  is a dimensionless magnetic field given by  $\hat{B} = B\delta_t/(\delta_x\sqrt{\rho_0\mu_0})$  where  $\rho_0$  is a reference density and  $\mu_0$  the magnetic permeability. Here,  $i \in \{x, y, z\}$  represents the Cartesian coordinate directions. Note that this particular nondimensionalisation is chosen so that the Alfvén velocity can be kept small by choosing an appropriate value of  $\rho_0$ . Due to the weakly compressible nature of the approach, this is necessary to avoid inaccuracies resulting from certain higher order terms [12].

The magnetic field can be recovered from the model by taking zeroth kinetic moments; its components are given by:

$$\hat{B}_i = \sum_{\alpha=0}^6 h_{\alpha i}. \quad (21)$$

The current density,  $\hat{J}_k$  can be calculated in two ways. One is from simply taking the curl of the magnetic field ( $\nabla \times \mathbf{B}$ ). The other, specific to kinetic approaches, is to extract it from the higher kinetic moments which avoids the need to take finite differences:

$$\hat{J}_k = -\frac{4}{\mu_0 c^2 \tau_m} \epsilon_{ijk} \sum_{\alpha=0}^6 (e_{\alpha i}^m h_{\alpha j} - e_{\alpha i}^m h_{\alpha j}^{\text{eq}}), \quad (22)$$

where

$$\sum_{\alpha=0}^6 e_{\alpha i}^m h_{\alpha j}^{\text{eq}} = (u_j \hat{B}_i - \hat{B}_j u_i), \quad (23)$$

$\hat{J}$  is a dimensionless current density given by  $\hat{J} = J\delta_t\sqrt{\mu_0/\rho_0}$  and  $\epsilon_{ijk}$  is the Levi–Civita permutation tensor.

This latter approach is more efficient in complex geometries, since calculations of finite differences can prove computationally time consuming for nodes located near boundaries, leading to a significant effect on the overall performance in certain types of geometry.

This formulation recovers the full form of the magnetic induction Eq. (16), which contrasts with many other procedures which solve a reduced form of the induction equation, neglecting higher order terms in the induced magnetic field. In many situations, the difference is not important, but for cases with high magnetic Reynolds number flows where the induced field is of the same order as the applied field, a reduced formulation is not appropriate.

As well as satisfying the induction equation, the field must also have zero divergence ( $\nabla \cdot \mathbf{B} = 0$ ) as required by Maxwell’s equations. It can be shown that provided the initial conditions have zero divergence, this requirement is automatically satisfied by this scheme [12].

This lattice Boltzmann formulation of the induction equation is accurate only in the limit of small Alfvén velocity. For higher velocities, errors from terms associated with certain higher order moments of the distribution function may become significant [12]. For this reason, the density  $\rho_0$  used to nondimensionalise the magnetic field should be selected such that  $\hat{B}$  remains small. In the current work, the reference density was selected such that the maximum value of the dimensionless applied field was 0.1. This is similar to the requirement that  $v \ll c_s$  for the LBM to reproduce the behaviour of the weakly compressible hydrodynamic equations.

Note that the LBM approach presented in Refs. [12,13] do not address the issue of simulating low magnetic Prandtl number flows at steady state or the specification of magnetic boundary conditions through distribution functions. Also, they do not consider the use of non-uniform grids required to resolve thin wall layers in high Hartmann number flows. These elements, which

are important for fusion applications, are presented in the following sections. Moreover, in contrast to that in [12,13], the present approach uses the MRT collision model for simulation of hydrodynamic fields with significantly enhanced numerical stability. This work also provides validations for a variety of benchmark problems relevant to fusion applications under different conditions.

#### 2.4. Low magnetic Prandtl number steady state magnetohydrodynamics

Most MHD flows of interest involve liquid metals and for these fluids there is a large difference between the timescales for the fluid flow and the magnetic induction. The ratio of the timescales is given by the magnetic Prandtl number  $Pr_m = \nu\sigma\mu_m$ , where  $\nu$  is the kinematic viscosity,  $\sigma$  is the electrical conductivity and  $\mu_m$  the permeability. For liquid metals,  $Pr_m \sim 10^{-7}$ , leading to a large disparity in time scales for the hydrodynamic and magnetic induction fields. Although the original LBM solution scheme performs well when the time scales are similar, it becomes numerically unstable when the time scales differ by a few orders of magnitude. This issue does not appear to have been addressed in previous works [12,13], but in this work the problem was resolved by rewriting the induction equation in the form

$$\frac{\partial \mathbf{B}}{\partial t} + \frac{\chi}{\gamma_m} \nabla \cdot (\mathbf{u}\mathbf{B} - \mathbf{B}\mathbf{u}) = \frac{1}{\gamma_m} \eta \nabla^2 \mathbf{B}, \quad (24)$$

which is appropriate for simulating steady state low  $Pr_m$  MHD flows.

The parameter  $\chi$  allows the selection of an effective magnetic resistivity ( $\eta = 1/(\sigma\mu)$ ) to fix an appropriate  $Pr_m$  and  $\gamma_m$  is a preconditioning parameter introduced to allow different time steps to be used to accelerate steady-state convergence. The use of this equation and the selection of  $\chi$  and  $\gamma_m$  is discussed in depth elsewhere [24]. In brief, it results in the following modifications to Eqs. (19), (20) and (22) in the approach presented in Section 2.3.

$$\tau'_m = \frac{1}{2} + \frac{4\eta}{\gamma_m} \frac{\delta_t}{(\delta_x)^2}, \quad (25)$$

$$h_{\alpha i}^{\text{eq}} = W_\alpha \left[ \hat{B}_i + \frac{4e_{\alpha j}^m}{c^2} \left( \frac{\chi}{\gamma_m} \right) (u_j \hat{B}_i - \hat{B}_j u_i) \right], \quad (26)$$

$$\hat{J}_k = -\frac{4}{\chi\mu_m c^2 \tau'_m} \varepsilon_{ijk} \sum_{\alpha=0}^6 (e_{\alpha i}^m h_{\alpha j} - e_{\alpha i}^m h_{\alpha j}^{\text{eq}}). \quad (27)$$

Another point to note is that this approach is only strictly valid for steady state flows, though in practice it can be applied to transient flows provided the timescale for the magnetic field remains much less than that for the fluid motion. A similar modification to the fluid flow equation can also be made to allow steady state solutions of both hydrodynamic and magnetic fields to be reached much more rapidly [24].

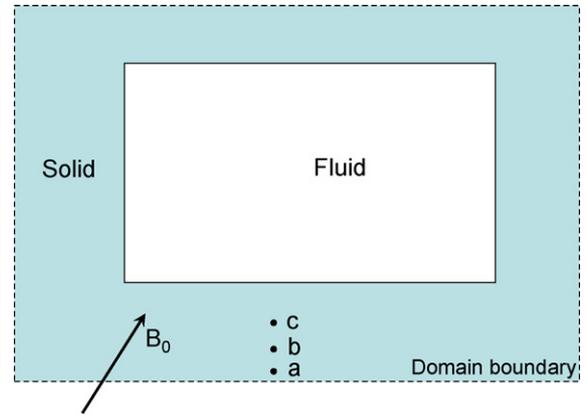


Fig. 4. Implementation of magnetic boundary condition in LBM. The magnetic boundary can, in general, be extended outside the fluid region so that the computational domain encompasses induced fields.

#### 2.5. Boundary conditions for magnetic induction fields

For the induction equation, no suitable formulations for the boundary conditions could be found in the literature. Earlier work [12] 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 technique for applying boundary conditions for the applied field was therefore formulated. This method enabled the magnetic field vector to be specified at the boundary, which is generally equivalent to imposing zero induced field. In practice, the induced magnetic field will extend outside the fluid region, and for this reason, greater accuracy may be obtained if the domain over which the magnetic induction equation is solved extends outside the fluid region. The approach taken was similar to the extrapolation scheme used for the fluid flow [21], but sets the magnetic field rather than the velocity to a specified value at the boundary. The procedure is as follows:

- for nodes on domain boundary (e.g. point a in Fig. 4), use the imposed field for the calculation of the equilibrium distribution in Eq. (20).
- use these values to calculate the post-collision distributions with Eq. (17),  $\tilde{h}_\alpha$  on boundary points.
- perform the advection step.
- for directions  $\alpha$  facing into the domain, the post-advection values of  $h_\alpha$  will be unknown. These unknown distributions are then calculated using the second-order extrapolation scheme.

$$h_\alpha(x, t + \delta_t) = 2\tilde{h}_\alpha(x, t) - \tilde{h}_\alpha(x + e_\alpha \delta_t, t), \quad (28)$$

where  $x$  and  $x + e_\alpha \delta_t$  correspond to points a and b in Fig. 4.

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 with the above. For reasons of numerical stability it was found that the two points neighbouring a face on which the magnetic field is set should have the same electrical conductivity; this may result in one more extra layer of grid points than might otherwise



Fig. 5. Schematic diagram of pipe flow with magnetic field applied perpendicular to flow.

have been needed. For example to impose an insulating wall boundary condition, in Fig. 4, the grid points a and b should both lie within the solid region (which in this case is assigned a very low conductivity). The insulating boundary will then lie midway between points b and c. Conducting walls can be handled in a similar way by selecting appropriate values for the conductivity for the wall.

### 3. Results and discussion: multidimensional MHD benchmark problems

#### 3.1. Laminar MHD canonical flows

Several standard test cases for which analytical solutions exist are available for the validation of computer codes. These include the flow between two parallel plates with a magnetic field imposed perpendicular to the plates (Hartmann flow), MHD flow in a two dimensional duct [25], and MHD flow through a pipe [26]. For more complex cases, such as 3D lid-driven cavity flow, where there are no analytical solutions, comparisons with results obtained with other numerical solution schemes can be made.

Preliminary tests with Hartmann flow gave excellent agreement with the analytical solution. However, since many of the terms in the governing field equations are zero in this case, it was necessary to also try more complex canonical problems for a comprehensive validation, and these are detailed in the following sections.

##### 3.1.1. MHD flow through a cylindrical pipe

The laminar flow through a pipe with a field applied normal to the flow, often known as the Gold problem [26], is illustrated in Fig. 5. No slip velocity boundary conditions through the approach presented in Section 2.2, and electrically insulating boundary conditions imposed through the approach presented in Section 2.4 are imposed.

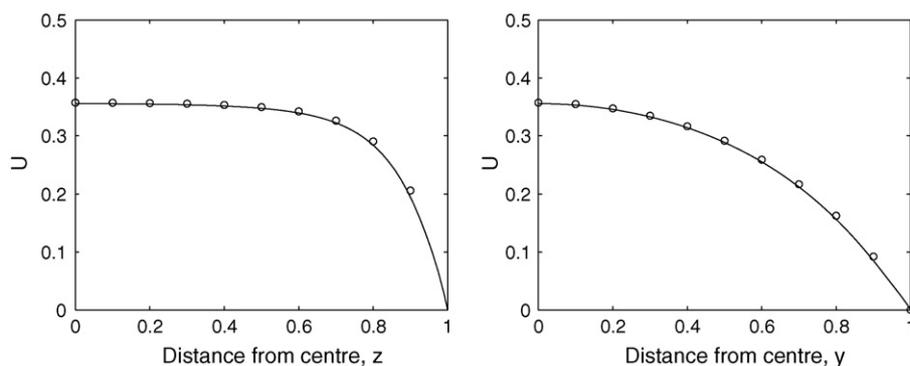


Fig. 6. Velocity profiles for directions parallel (left) and perpendicular (right) to applied field for MHD pipe flow with  $Ha = 10$ . Line is LBM predictions and circles are analytical solution.

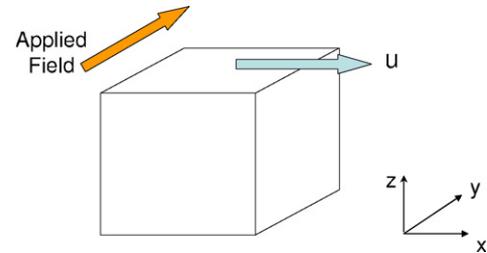


Fig. 7. Schematic view of lid-driven cavity flow problem.

Fig. 6 shows the velocity profiles for the directions parallel and perpendicular to the imposed magnetic field,  $B_0$  for a Hartmann number ( $Ha = B_0 r / \sqrt{\sigma / \rho \nu}$ ) of 10 (where the length scale  $r$  is taken to be the pipe radius). Velocities are normalised with the peak velocity for the case with no applied field, and the distance with the pipe radius. In this simulation, 120 points were used to resolve the diameter of the pipe. The analytical solution was calculated using the formulation given in [26] and good agreement can be seen.

##### 3.1.2. 3-D MHD lid-driven cavity flow

A true 3-D MHD validation case is provided by the lid-driven cavity flow, which sets up complex fluid flow that is strongly influenced by the applied magnetic field in all coordinate directions. In the case presented here, the flow is in a cubic box with a the top lid moving in the  $x$  direction and the magnetic field applied in the  $y$  direction as in Fig. 7. The Reynolds number was 100 and the Hartmann number was 45; both characteristic numbers were calculated using the side of the cube as the reference length.

Fig. 8 compares the results with those obtained with a finite difference code [3]; an analytical solution is not available for this problem. 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 (uniform grid,  $128^3$ ), and examination of runs with different resolutions upheld this viewpoint.

Due to the discontinuity at the edges, this is a demanding problem, and the range of time steps that could be used with the more common single relaxation time (SRT) collision model without the computation becoming unstable was very restrictive.

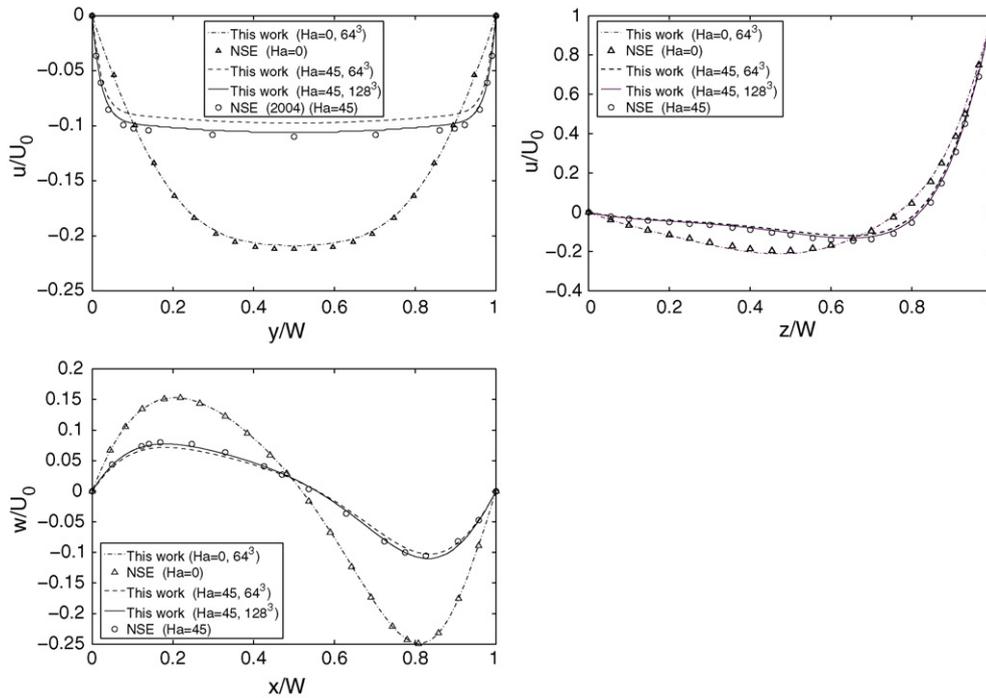


Fig. 8. Comparison of results of lid-driven cavity problem with those of a finite difference code involving direct solution of the Navier–Stokes equations (NSE) [3].  $Ha=45$ ,  $Re=100$ .

The SRT model uses a single dimensionless relaxation time,  $\tau = 0.5 + 3\nu\delta_t/\delta_x^2$  in the collision term, (here  $\nu$  is the kinematic viscosity) and it was necessary for  $\tau$  to be very close to, or greater than unity. However, with the MRT model for the hydrodynamics the code was found to be much more stable, and a value of  $\tau$  (which forms a component of the relaxation matrix in the MRT model) as low as about 0.55 at  $Ha=45$  could be used. This corresponds to a factor of 10 in the minimum viscosity that can be used compared with the SRT model, or an order of magnitude increase in the Reynolds number that could be simulated. Note that in both cases, the induction equation was solved using an SRT model.

The rate at which the solution converges to its steady-state value is shown in Fig. 9. In this case, the viscosity and resistivity were set to be equal, and a flow with an effective Prandtl number of  $5.625 \times 10^{-7}$  was simulated by setting  $\chi$  to that value. The variable  $\gamma_m$  is the preconditioning parameter appearing in the magnetic induction equation and  $\gamma$  is the corresponding parameter for hydrodynamics. More details on the implementation of this method are given in Ref. [24]. Two values of  $\gamma_m$  are shown in the figure, and the rate at which the solution converges is substantially increased by using a lower value of  $\gamma_m$ . Note that  $\gamma_m = \gamma = 1$  corresponds to the case in which no preconditioning is applied.

### 3.2. Turbulent MHD flow

Let us now consider the simulation of turbulent flow in the presence of a magnetic field through a cylindrical pipe as illustrated earlier in Fig. 5. A fully-developed turbulent flow entered a pipe; at the inlet, there was no magnetic field, but farther

downstream it entered a region where a field was imposed perpendicular to the pipe axis. In the LBM computations, 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 [27]. Statistics were collected downstream to see how the field had affected the turbulence.

The inlet conditions were obtained by running a pipe flow simulation with periodic boundary conditions and a body force in the axial direction. After a statistically steady state had been reached, the flow field on one cross sectional plane was collected every time step for 5000 time steps. This set of velocity fields was used for the inlet conditions and was recycled every 10,000

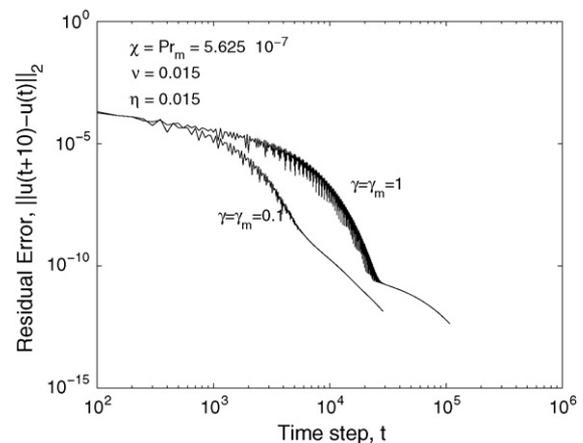


Fig. 9. Convergence to steady state for 3-D lid driven cavity. Residual error is a measure of the rms velocity difference between iterations separated by 10 time steps, and  $\gamma$  and  $\gamma_m$  are speed-up parameters in hydrodynamic and MHD equations respectively and  $\gamma=1$  corresponds to no convergence acceleration.

time steps. The Reynolds number based on the bulk velocity and the diameter was 4900.

Unresolved turbulent scales were handled by using a Smagorinsky subgrid scale (SGS) eddy viscosity model with van Driest wall damping [11]. The presence of MHD forces is known to affect turbulence, and the Smagorinsky formulation can be modified to reflect this [28]; for the conditions used in this case, these corrections were found to be very small, so the standard Smagorinsky model was used. The computations were run on the US Department of Energy's Bassi supercomputer, housed at NERSC. Bassi is a distributed memory computer with 888 IBM p575 processors running at 1.9 GHz, each with 4 GBytes of memory. The domain size used for the large eddy simulation (LES) was  $2400 \times 120 \times 120$  and this was run on 32 processors, which resulted in the computation proceeding at the rate of about 1500 time steps per hour. With this domain size, a length of 20 diameters was simulated, and Fig. 10 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.

For the case with no MHD, it can be seen that the profiles of turbulence intensities predicted by the LBM are similar to those from a DNS [29], though there is some overprediction of turbulence, particularly for the streamwise direction. Studies showed this could be attributed to the resolution used, with the error changing roughly as the square of the grid spacing. Also, LES generally overpredicts the turbulence intensity in the streamwise direction.

At 15 diameters downstream, all three components of turbulent fluctuations are substantially reduced, as is expected for an MHD flow where the field is applied in the wall-normal direction. Another feature is that the peaks in the intensities are shifted away from the wall, an effect that was also found by [30].

A set of tests using different numbers of processors was undertaken for this problem to assess the parallel performance of the code. The code was run for a short duration and the elapsed time taken to perform a fixed number of time steps was recorded. In all the tests the same parameters and grid resolution were used, the only difference being in the number of processors. In each case a period of 50 time steps was used, and to avoid any effects from the initialisation of the program the code was run for 50 time steps before starting the timed period. Fig. 11 shows the speed of the computation for different numbers of processors. The largest case used 128 processors and the smallest 16; memory constraints prevented tests with fewer processors. The speed is defined as the reciprocal of the elapsed time, and is normalised such the case with 16 processors has a speed of 16. This case actually took 4.6 s to complete each time step. The speed up can be seen to be almost linear, with the 128 processor case reaching a speed of 113 on this scale – a drop of only about 12% below the value for a linear variation.

Slab decomposition along the longest axis was used for these tests, with ghost points used at each face of the subdomains to store information exchanged with other processors. For 16 processors, the largest subdomain would have been 152 grid points wide and for the 128 processor case 21 points wide (19 active and 2 ghost points). Although the ghost points do not

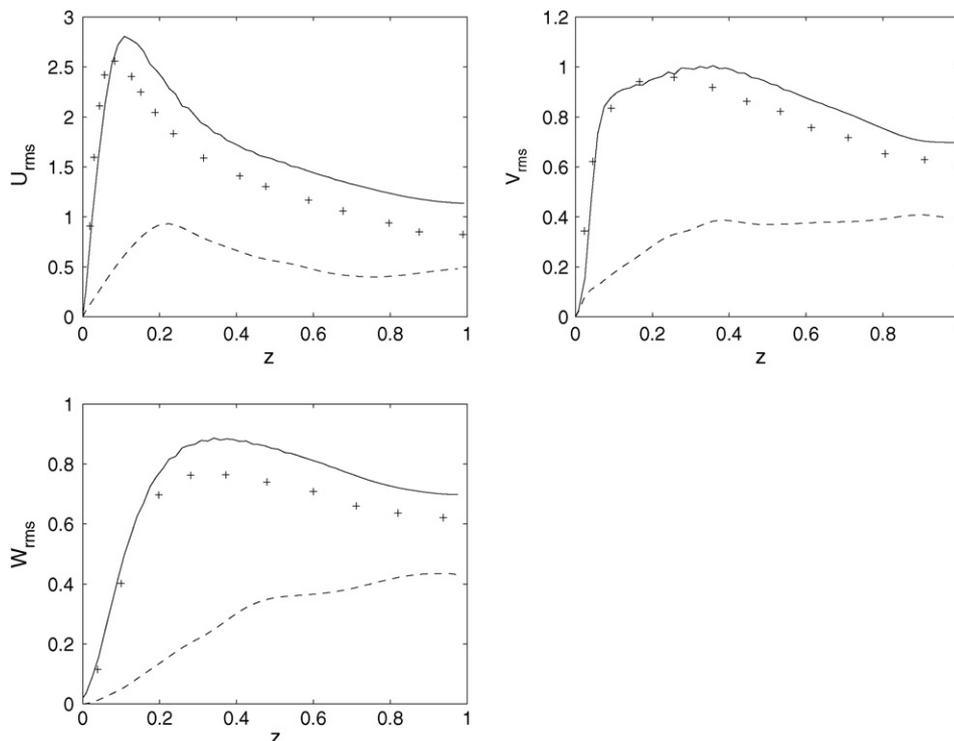


Fig. 10. Profiles of rms velocity fluctuations along  $z$  axis through centre, taken 15 diameters from entry into the magnetic field. Solid line is LBM for no magnetic field ( $Ha=0$ ), crosses DNS [29] at  $Re=5300, Ha=0$  and dashed line LBM with  $Ha=20$ .

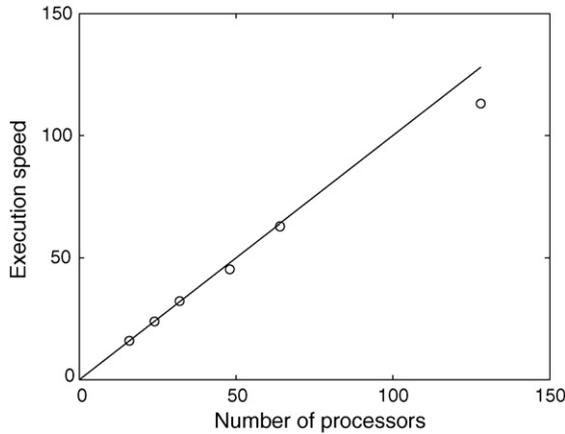


Fig. 11. Speed of computations for MHD turbulent pipe flow, on NERSC Bassi supercomputer. Circles are data points and solid line corresponds to linear scaling. Grid size was  $2400 \times 120 \times 120$  and speeds are normalised to give 16 processor case a speed of 16.

perform all the computational steps involved in the LBM, the extra computational workload probably accounts for a substantial proportion of the 12% fall found. Other sources would be from the time involved in exchanging data and the variations in speed of different nodes on the supercomputer. This latter one probably accounts for a few percent – when running smaller computations on single nodes of the Bassi machine (eight processors) execution times found to vary by up to about 5% from one node to another.

#### 4. High Hartmann number MHD flows using interpolated-streaming LBM approach

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, which are important in fusion applications, 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. For example, the thickness of Hartmann layers, (perpendicular to applied field) varies as  $\delta_H \propto 1/Ha$  while for the so-called side layers (parallel to applied field) the variation goes as  $\delta_H \propto 1/\sqrt{Ha}$  [22].

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) [31] 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 [32]. Thus, it was proposed by [33] that the collision step can be computed locally on the lattice grids in the

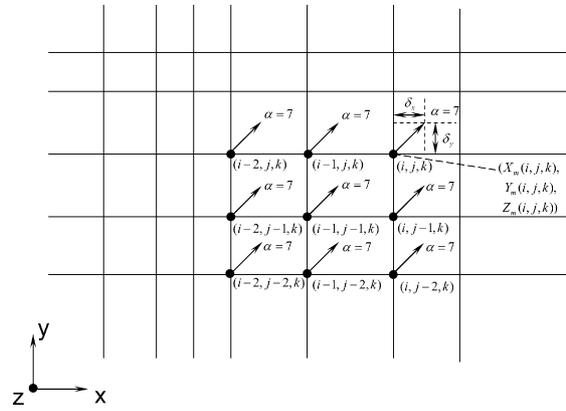


Fig. 12. Illustration of a second-order upwind Lagrangian interpolation for a particle velocity direction ( $\alpha = 7$ ) to implement non-uniform grids in the LBM.

usual manner; after the collision step, the particle distributions move according to their velocities  $e_\alpha$ , although the advected distance of the particle distributions may not, in general, coincide with the mesh spacing (as shown in the example in Fig. 12). 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 [34]. Thus, the order of interpolation schemes should not be lower than the order of accuracy of the LBE. This approach is often termed the interpolation-supplemented lattice Boltzmann method (ISLBM). It has been applied to simulate a variety of problems, for example to simulate turbulent plane jets using a two-equation turbulent modelling approach within the LBM framework [35].

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-LBE and the magnetic induction equation, and this is indeed what was done in this work. A second-order Lagrange interpolation was employed in the implementation, which is 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 A.

With this modification, it allows 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. Moreover, it is relatively easy to implement and naturally amenable for parallel implementation.

##### 4.1. 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  $B_i$  [22]

$$v(z) = \left(-\frac{dP}{dx}\right) \frac{L}{B_0} \sqrt{\frac{\rho}{\sigma\nu}} \left[ \frac{1 - \cosh(Ha z/L)/\cosh(Ha)}{\tanh(Ha)} \right] \tag{29}$$

$$B_i(z) = \left(-\frac{dP}{dx}\right) \frac{\mu_0 \rho L}{B_0} [\sinh(Ha z/L)/\sinh(Ha) - z/L] \tag{30}$$

where  $-(dP/dx)$  is the driving pressure gradient,  $\sigma$  the electrical conductivity,  $\nu$  the kinematic viscosity and  $\mu_0$  is the 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:

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

The solution is discussed in depth in Ref. [23], where the more general solution for finite conductivity walls and a square duct is also given. Note that at very high Hartmann numbers, it is difficult to evaluate the above analytical solution directly, since the terms inside the hyperbolic functions become too large. In this situation, approximate (asymptotic) forms for high  $Ha$  can be used (this is also addressed by Ref. [23]).

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 the Roberts transformation [36] in which a uniform grid (i.e. with equally spaced points)  $\bar{z}_k$  in the range  $0 < \bar{z}_k < 1$  is transformed into a stretched grid  $z$ :

$$z = 2L \frac{(\beta + 2\alpha)[(\beta + 1)/(\beta - 1)]^{(\bar{z}-\alpha)/(1-\alpha)} - \beta + 2\alpha}{(2\alpha + 1)\{1 + [(\beta + 1)/(\beta - 1)]^{(\bar{z}-\alpha)/(1-\alpha)}\}} \tag{31}$$

where the parameter  $\alpha$  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  $\beta$  determines the degree of stretching, and Smolentsev et al. [37] suggest using a value

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

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  $\sim 1/Ha$ . Inspection of profiles generated suggested that the boundary layer thickness,  $\delta/L$ , was actually  $\sim 5/Ha$ , 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:

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

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

Figs. 13–15 show the velocity and induced magnetic field profiles for Hartmann numbers of 100, 1000, and 10,000, i.e. spanning two-orders of magnitude. The coordinate  $z$  in these figures runs from 0 at the wall to 1 at the centre. Fig. 13 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. 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 dramatic saving in computer resources (less than 0.5% of the computational effort is required when the ISLBM is used in lieu of the uniform grid LBM).

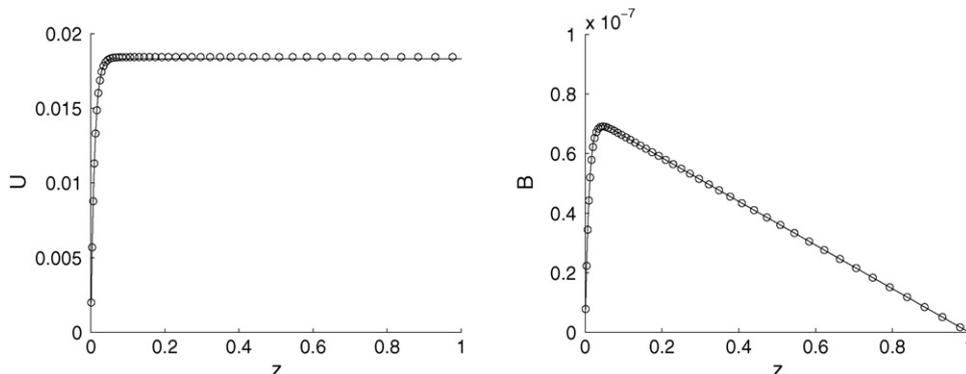


Fig. 13. Velocity ( $U$ ) and magnetic field ( $B$ ) profiles for Hartmann flow between parallel plates at  $Ha = 100$ ; line LBM, circles analytical. 96 grid points.

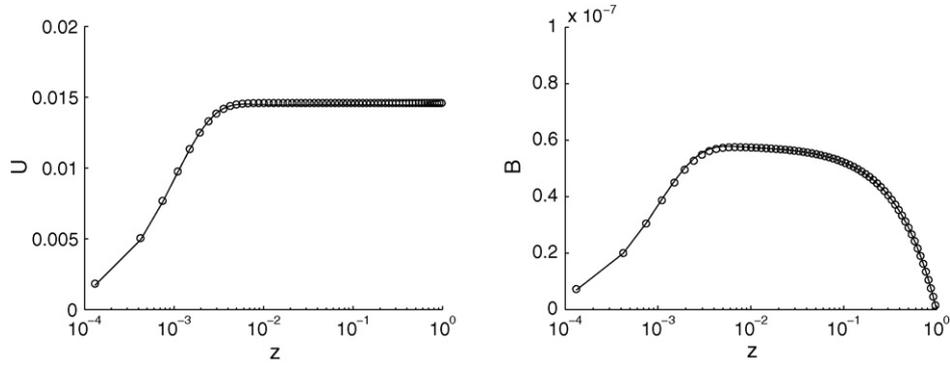


Fig. 14. Velocity and magnetic field for Hartmann flow between parallel plates at  $Ha = 1000$ ; line LBM, circles analytical. 128 grid points. Log scale used for abscissae.

4.2. 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  $1 = Ha$ , but for those parallel to the field (the *side* layers), the width varies as  $1/\sqrt{Ha}$ . The case was resolved by using 128 grid points in each direction, and Fig. 16 shows the velocity profiles obtained with the LBM along the wall bisectors along with the analytical solutions. The veloc-

ity 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 distributions 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.

The profile through the side walls was calculated with the following approximate asymptotic formula [22]

$$u = u_0(1 + \eta^2)(1 - \text{erf}(\eta/\sqrt{2})) - \eta\sqrt{2/\pi} \exp(-\eta^2/2) \quad (32)$$

where  $\eta$  is a dimensionless distance  $\eta = y\sqrt{Ha}/2$ , with  $y$  being the distance from the wall, normalised with the duct half width.

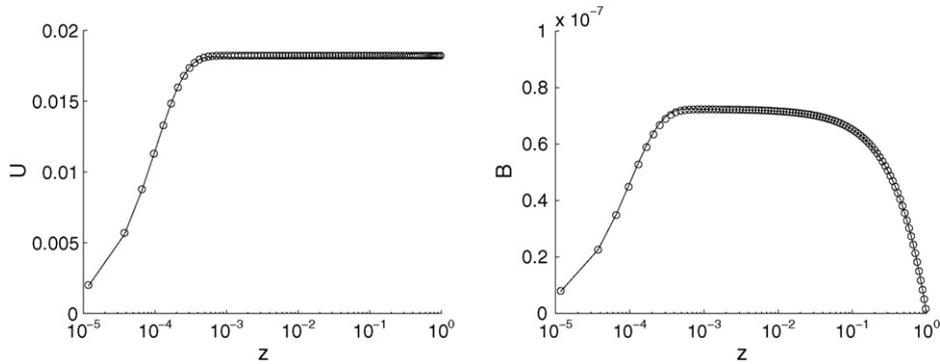


Fig. 15. Velocity and magnetic field for Hartmann flow between parallel plates at  $Ha = 10000$ ; line LBM, circles analytical. 192 grid points. Log scale used for abscissae.

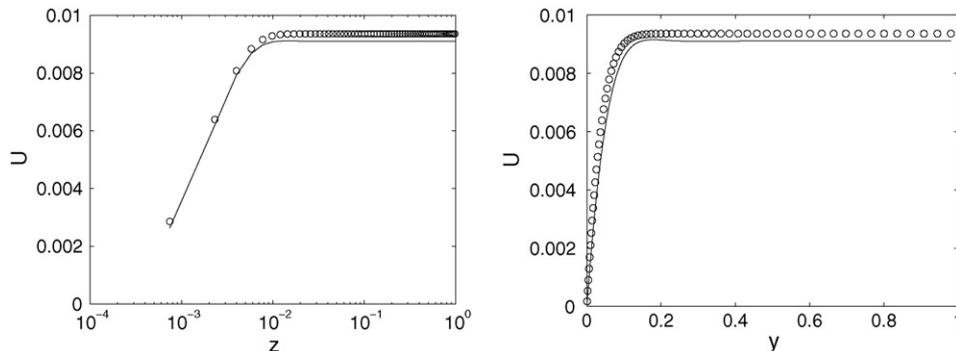


Fig. 16. Velocity profiles along wall bisectors for MHD flow in an insulating duct at  $Ha = 500$ , left: side layers and right: Hartmann layer; line LBM, circles analytical solution. Field applied in  $z$  direction. Note that for clarity the right graph has log scale for the abscissa, and axes are normalised by the duct half-width.

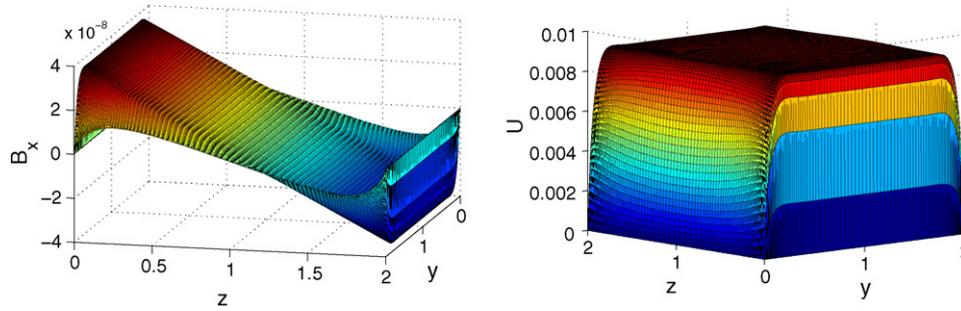


Fig. 17. Plots of induced magnetic field ( $B_x$ ) and velocity ( $U$ ) for MHD flow in an insulating duct at  $Ha=500$ .

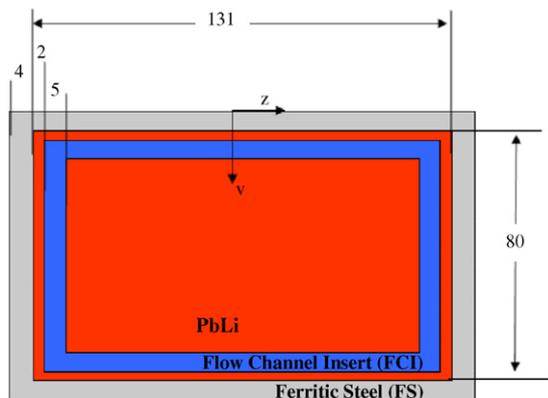
The velocity  $u_0$  is the velocity at the centre for the equivalent case with no side walls. This solution is also given a thorough discussion by Ref. [23], but readers are warned that during the course of this study, the equation given in their book was found to contain a typographical error.

Two dimensional plots of magnetic field and the velocity field are shown in Fig. 17. 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.

It was found that although extreme amounts of stretching (factors  $> 100$ ) work perfectly well in regions where the flow parameters do not change very little, such as in the core of Hartmann flow, more moderate stretching values are required in other regions. With large values of the stretching factor in the side or Hartmann layers, the computation can lose accuracy and problems with convergence to a steady state may arise.

### 5. Fusion MHD example: 3D simulation of MHD flow in a thermal blanket module

One case undertaken that is relevant to a current fusion engineering problem was the simulation of a section of a thermal blanket. A cross-section of the geometry for the problem tackled is shown in Fig. 18. The widths of the layers are assumed to be uniform around the perimeter, and the



All dimensions in mm, not to scale

Fig. 18. Schematic diagram showing cross-section of thermal blanket module.

cell is 1.4 m 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  $0.1 \text{ ms}^{-1}$ . The magnetic field is applied in the  $z$  direction. 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 transformation, as discussed before, 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.

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 fluctuations 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 method used to accelerate convergence to steady state. Very similar behaviour has previously been observed in a this sort of problem with a finite volume method [37]; 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 higher resolution grids and with the use of multi-processor computer clusters.

Fig. 19 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 duct flows with highly conducting walls, these peaks can become very narrow, and are known as side wall jets. In the other direction, there is a very steep climb in

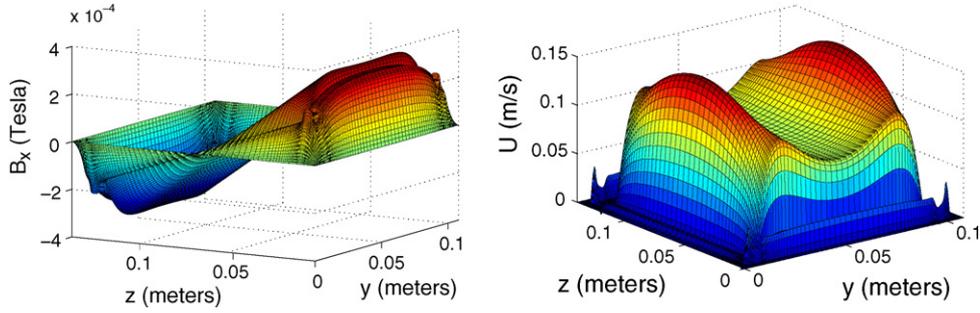


Fig. 19. Induced field ( $B_x$ ) and velocity ( $U$ ) plots on a cross-section of a thermal blanket module at  $Ha = 100$ .

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. The profiles of the velocity induced field are consistent with expectations.

## 6. Summary and conclusions

A number of new developments have been introduced for the simulation of MHD flows with the lattice Boltzmann method. The main ones were:

- introduction of a new method for specifying the magnetic field at a boundary, through an extrapolation scheme for the vector distribution function. Unlike earlier LBM approaches, there is no restriction on the angle of the field relative to the coordinate axes.
- a new technique, based on a preconditioning approach, for allowing simulations of MHD flows with low magnetic Prandtl numbers and for steady-state convergence.
- non-uniform gridding approach to resolve thin layers near the walls at high Hartmann numbers developed through an interpolated-streaming step in the LBM for both scalar (hydrodynamics) and vector (magnetic induction) distribution functions.
- introduction of an advanced multiple relaxation time (MRT) collision model to solve MHD problems using the LBM for enhanced fidelity and numerical stability.

solutions have shown very good agreement. For the more complex benchmarks, such as the 3-D lid-driven cavity flow, analytical solutions are not available, but comparisons with predictions of other CFD codes showed close agreement with benchmark results, verifying the validity of the new advances. For many problems, the use of the MRT model in lieu of the SRT model allows an improvement in numerical stability by a factor of about 3, and for particularly demanding problems, such as the MHD lid driven cavity, an order of magnitude improvement can sometimes be seen.

The extension of the ISLBM to MHD flows allows flows with Hartmann numbers of several thousand to be simulated relatively quickly. By appropriately selecting the distribution and clustering of grid points, it has been found that the computer time requirement can be reduced for a factor of several hundred, when compared with that of uniform grids. Also, the LBM has demonstrated very good parallel performance on multi-processor machines, with tests showing almost linear scaling up to 128 processors for evaluated MHD problems.

Designs of heat transport systems in fusion reactors typically involve 3-D bounded liquid metals flows with low magnetic Prandtl number ( $Pr_m = \nu/\eta_m \sim 10^{-7}$ ). Magnetic fields are generally high, leading to narrow shear layers near the walls which require the use of stretched grids for accurate and computationally efficient solutions. With the new developments discussed here, the LBM is now in a position to simulate actual systems for proposed fusion reactors, and a preliminary example of such a computation for a thermal blanket is indeed presented in this paper. The ability of the LBM to parallelise efficiently will enable much larger problems to be tackled than would otherwise be the case (see Fig. 12).

## Appendix A

$$\begin{aligned}
 f_{7,i,j,k} = & a_{1,i,j,k}^- [b_{1,i,j,k}^- \tilde{f}_{7,i,j,k} + b_{2,i,j-1,k}^- \tilde{f}_{7,i,j-1,k} + b_{3,i,j-2,k}^- \tilde{f}_{7,i,j-2,k}] \\
 & + a_{2,i,j,k}^- [b_{1,i-1,j,k}^- \tilde{f}_{7,i-1,j,k} + b_{2,i-1,j-1,k}^- \tilde{f}_{7,i-1,j-1,k} + b_{3,i-1,j-2,k}^- \tilde{f}_{7,i-1,j-2,k}] \\
 & + a_{3,i,j,k}^- [b_{1,i-2,j,k}^- \tilde{f}_{7,i-2,j,k} + b_{2,i-2,j-1,k}^- \tilde{f}_{7,i-2,j-1,k} + b_{3,i-2,j-2,k}^- \tilde{f}_{7,i-2,j-2,k}]
 \end{aligned} \tag{A.1}$$

A comprehensive set of validations and tests has been undertaken including both laminar and turbulent MHD flows. For the simpler canonical problems, comparisons with analytical

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 as in Eq. (3-a). The upwind

Lagrange interpolation coefficients in Eq. (A.1) are given below:

$$a_{1,i,j,k}^- = \left[ \frac{X_{m(i,j,k)} - (X_{m(i-1,j,k)} + \delta x)}{(X_{m(i,j,k)} + \delta x) - (X_{m(i-1,j,k)} + \delta x)} \right] \times \left[ \frac{X_{m(i,j,k)} - (X_{m(i-2,j,k)} + \delta x)}{(X_{m(i,j,k)} + \delta x) - (X_{m(i-2,j,k)} + \delta x)} \right] \quad (\text{A.2})$$

$$a_{2,i,j,k}^- = \left[ \frac{X_{m(i,j,k)} - (X_{m(i,j,k)} + \delta x)}{(X_{m(i-1,j,k)} + \delta x) - (X_{m(i,j,k)} + \delta x)} \right] \times \left[ \frac{X_{m(i,j,k)} - (X_{m(i-2,j,k)} + \delta x)}{(X_{m(i-1,j,k)} + \delta x) - (X_{m(i-2,j,k)} + \delta x)} \right] \quad (\text{A.3})$$

$$a_{3,i,j,k}^- = \left[ \frac{X_{m(i,j,k)} - (X_{m(i,j,k)} + \delta x)}{(X_{m(i-2,j,k)} + \delta x) - (X_{m(i,j,k)} + \delta x)} \right] \times \left[ \frac{X_{m(i,j,k)} - (X_{m(i-1,j,k)} + \delta x)}{(X_{m(i-2,j,k)} + \delta x) - (X_{m(i-1,j,k)} + \delta x)} \right] \quad (\text{A.4})$$

and

$$b_{1,i,j,k}^- = \left[ \frac{Y_{m(i,j,k)} - (Y_{m(i,j-1,k)} + \delta y)}{(Y_{m(i,j,k)} + \delta y) - (Y_{m(i,j-1,k)} + \delta y)} \right] \times \left[ \frac{Y_{m(i,j,k)} - (Y_{m(i,j-2,k)} + \delta y)}{(Y_{m(i,j,k)} + \delta y) - (Y_{m(i,j-2,k)} + \delta y)} \right] \quad (\text{A.5})$$

$$b_{2,i,j,k}^- = \left[ \frac{Y_{m(i,j,k)} - (Y_{m(i,j-1,k)} + \delta y)}{(Y_{m(i,j-1,k)} + \delta y) - (Y_{m(i,j,k)} + \delta y)} \right] \times \left[ \frac{Y_{m(i,j,k)} - (Y_{m(i,j-2,k)} + \delta y)}{(Y_{m(i,j-1,k)} + \delta y) - (Y_{m(i,j-2,k)} + \delta y)} \right] \quad (\text{A.6})$$

$$b_{3,i,j,k}^- = \left[ \frac{Y_{m(i,j,k)} - (Y_{m(i,j,k)} + \delta y)}{(Y_{m(i,j-2,k)} + \delta y) - (Y_{m(i,j,k)} + \delta y)} \right] \times \left[ \frac{Y_{m(i,j,k)} - (Y_{m(i,j-2,k)} + \delta y)}{(Y_{m(i,j-2,k)} + \delta y) - (Y_{m(i,j-1,k)} + \delta y)} \right] \quad (\text{A.7})$$

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,  $\delta x$ ,  $\delta y$  and  $\delta z$ , are the streaming distances based on the particle velocity  $c$  during a time interval  $\delta t$ , 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. 1), 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  $b_{3,i,j,k}^+$ . This approach can be extended to vector distribution functions using similar interpolation formulae.

## References

- [1] S. Smolentsev, N.B. Morley, M.A. Abdou, R. Munipalli, R. Moreau, Current approaches to modeling MHD flows in the dual coolant lead lithium blanket, *Magnetohydrodynamics* 42 (2006) 225–236.
- [2] N.B. Morley, S. Smolentsev, L. Barleon, I.R. Kirillov, M. Takahashi, Liquid magnetohydrodynamics – recent progress and future directions for fusion, *Fusion Eng. Des.* 51–52 (2000) 701–713.
- [3] N.B. Morley, S. Smolentsev, R. Munipalli, M. Ni, D. Gao, M. Abdou, Progress on the modeling of liquid metal, free surface, MHD flows for fusion liquid walls, *Fusion Eng. Des.* 72 (2004) 3–34.
- [4] M.A. Abdou, A. Ying, N.B. Morley, et al., On the exploration of innovative concepts for fusion chamber technology, APEX Interim Report Overview, *Fusion Eng. Des.* 73 (2001) 83–93.
- [5] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* 30 (1998) 329–364.
- [6] P. Bhatnagar, M. Gross, E. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (1954) 511–525.
- [7] Y.H. Qian, D. d’Humières, P. Lallemand, Lattice BGK models for the Navier–Stokes equation, *Europhys. Lett.* 17 (1992) 479–484.
- [8] H. Chen, S. Chen, W.H. Matthaeus, Recovery of the Navier–Stokes equations using a lattice-gas Boltzmann method, *Phys. Rev. A* 45 (1992) 5339–5342.
- [9] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L.-S. Luo, Multiple-relaxation-time lattice Boltzmann models in three-dimensions, *Phil. Trans. R. Soc. Lond. A* 360 (2002) 437–451.
- [10] K.N. Premnath, J. Abraham, Three dimensional multi-relaxation-time (MRT) lattice Boltzmann models for multiphase flow, *J. Comput. Phys.* 224 (2007) 539–559.
- [11] K.N. Premnath, M.J. Pattison, S. Banerjee, Generalized lattice Boltzmann equation with forcing term for LES of bounded turbulent flows: accuracy, stability and computational efficiency, *Phys. Rev. E* (2007) (submitted for publication).
- [12] P.J. Dellar, Lattice kinetic schemes for magnetohydrodynamics, *J. Comp. Phys.* 179 (2002) 95–126.
- [13] G. Breyiannis, D. Valougeorgis, Lattice kinetic simulations in three-dimensional magnetohydrodynamics, *Phys. Rev. E* 69 (2004) 65–72.
- [14] J. Carter, M. Soe, L. Oliker, Y. Tsuda, G. Vahala, L. Vahala, A. Macnab, Magnetohydrodynamic Turbulence Simulations on the Earth Simulator Using the Lattice Boltzmann Method, in: B.B. Jeff Kuehn (Ed.), *Conference on High Performance Computing, Networking and Storage*, IEEE Computer Society Press, Seattle, WA, 2005.
- [15] R.K. Madabhushi, S.P. Vanka, Large eddy simulation of turbulence-driven secondary flow in a square duct, *Phys. Fluids A* 3 (1991) 2734–2745.
- [16] R. Mei, L.-S. Luo, W. Shyy, An accurate curved boundary treatment in the lattice Boltzmann method, *J. Comp. Phys.* 155 (1999) 307–330.
- [17] Z. Guo, C. Zheng, An extrapolation method for boundary conditions in lattice Boltzmann method, *Phys. Fluids* 14 (2002) 2007–2010.
- [18] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, *Phys. Fluids* 13 (2001) 3452–3459.
- [19] P. Lallemand, L.-S. Luo, Lattice Boltzmann method for moving boundaries, *J. Comp. Phys.* 184 (2003) 406–421.
- [20] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part I. Theoretical foundation, *J. Fluid Mech.* 271 (1994) 285–310.
- [21] S. Chen, D. Martinez, R. Mei, On boundary conditions in lattice Boltzmann methods, *Phys. Fluids* 8 (1996) 2527–2536.
- [22] R. Moreau, *Magnetohydrodynamics*, Kluwer Academic Publications, Dordrecht, Netherlands, 1990.
- [23] U. Müller, L. Bühler, *Magnetofluidynamics in Channels and Containers*, Springer-Verlag, Berlin, 2001.
- [24] K.N. Premnath, M.J. Pattison, Convergence acceleration of generalized lattice Boltzmann equation with forcing term through preconditioning for steady state flows and some applications, *Phys. Rev. E* (2007) (submitted for publication).

- [25] J.C.R. Hunt, Magnetohydrodynamic flow in rectangular ducts, *J. Fluid Mech.* 21 (1965) 577–590.
- [26] R.R. Gold, Magnetohydrodynamic pipe flow. Part 1, *J. Fluid Mech.* 13 (1962) 505–512.
- [27] T. Yokomine, J. Takeuchi, H. Nakaharai, S. Satake, T. Kunugi, N.B. Morley, M.A. Abdou, Experimental investigation of turbulent heat transfer of high Prandtl number fluid flow under strong magnetic field, *Fusion Sci. Tech.* 52 (2007) 625–629.
- [28] Y. Shimomura, Large eddy simulation of magnetohydrodynamic turbulent channel flows under a uniform magnetic field, *Phys. Fluids A* 3 (1991) 3098–3106.
- [29] J.G.M. Eggels, F. Unger, M.H. Weiss, J. Westerweel, R.J. Adrian, R. Friedrich, F.T.M. Nieuwstadt, Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment, *J. Fluid Mech.* 268 (1994) 175–209.
- [30] D. Lee, H. Choi, Magnetohydrodynamic turbulent flow in a channel at low magnetic Reynolds number, *J. Fluid Mech.* 439 (2001) 367–394.
- [31] D. Wolf-Gladrow, *Lattice-gas Cellular Automata and Lattice Boltzmann Models*, Lecture Notes in Mathematics, vol. 1725, Springer, New York, 2000.
- [32] N. Cao, S. Chen, S. Jin, D. Martinez, Physical symmetry and lattice symmetry in the lattice Boltzmann method, *Phys. Rev. E* 5 (1997) R21–R24.
- [33] X. He, L.-S. Luo, M. Dembo, Some progress in the lattice Boltzmann method. Part 1: Non-uniform mesh grids, *J. Comp. Phys.* 129 (1996) 357–363.
- [34] X. He, Error analysis for the interpolation-supplemented lattice-Boltzmann equation scheme, *Int. J. Mod. Phys. C* 8 (1997) 737–745.
- [35] K.N. Premnath, J. Abraham, Discrete lattice BGK equation computations of transient incompressible turbulent jets, *Int. J. Mod. Phys. C* 15 (2004) 699–719.
- [36] G.O. Roberts, Computational meshes for boundary layer problems *Proc. Second Int. Conf. Numerical Methods Fluid Dynamics*, Lect. Notes Phys. 8 (1971) 171–177.
- [37] S. Smolentsev, N.B. Morley, M.A. Abdou, Code development for analysis of MHD pressure drop in a liquid blanket using insulation technique based on a fully developed flow model, *Fusion Eng. Des.* 73 (2005) 83–93.