Available online at www.sciencedirect.com
SCIENCE DIRECT®
Physica A 362 (2006) 48–56
PHYSICA A
www.elsevier.com/locate/physa

Non-local closure and parallel performance of lattice Boltzmann models for some plasma physics problems

Angus I.D. Macnab[a,*], George Vahala[b], Linda Vahala[c], Jonathan Carter[d], Min Soe[e], William Dorland[f]

[a]CSCAMM, University of Maryland, 4119 CSIC Building, College Park, MD 20742, USA
[b]Department of Physics, College of William and Mary, P.O. Box 8795, Williamsburg, VA 23187-8795, USA
[c]Department of Electrical and Computer Engineering, Old Dominion University, 231 E Kaufman Hall, Norfolk, VA 23529, USA
[d]NERSC, Lawrence Berkeley National Laboratory, Berkeley, CA 94270, USA
[e]Department of Mathematics and Physics, Rogers State University, Claremore, OK 74017, USA
[f]Department of Physics, University of Maryland, College Park, MD 20742, USA

Available online 7 October 2005

Abstract

The lattice Boltzmann (LB) method is a mesoscopic approach to solving nonlinear macroscopic conservation equations. Because the LB algorithm yields a simple collide-stream sequence it has been extensively applied to Navier–Stokes flows, but its MHD counterpart is less well known in the plasma physics community. Several plasma problems that should be amenable to LB are discussed. In particular, Landau damping—a collisionless kinetic phenomenon of wave–particle interaction—can be studied by LB since non-local macroscopic closures have been generated by plasma physicists. The parallel performance of 2D LB codes for MHD are presented, including scaling performance on the Earth Simulator.

Keywords: Lattice Boltzmann; Magnetohydrodynamics; Landau damping

1. Introduction

Simulations of turbulent flows in complex geometry place great strain on computational algorithms designed for the direct solution of the Navier–Stokes and magnetohydrodynamic (MHD) equations. These algorithms require sophisticated schemes (high-order finite elements or Newton–Krylov solvers) to resolve the nonlinear convective derivatives with second-order accuracy and in a numerically stable manner. MHD has to handle even more nonlinear convective terms than fluid dynamics schemes. The lattice Boltzmann (LB) BGK approach [1–3] attempts to circumvent these nonlinear convective terms by differencing a set of distribution functions at first order to produce a second order solution in the space of the macroscopic variables. This is achieved by introducing an additional minimal set of discrete kinetic velocity lattice vectors. Because of the ease of handling complex boundaries [4] and its simple structure, which permits highly efficient parallelization

*Corresponding author.
E-mail addresses: amacnab@mailaps.org, amacnab@u.washington.edu (A.I.D. Macnab).

0378-4371/$ - see front matter
doi:10.1016/j.physa.2005.09.018

and vectorization [5], the LB method is becoming a competing tool for incompressible Navier–Stokes problems [6]. Following the advent of lattice gas models for MHD [7,8], a number of ingenious LB methods [1,9–11] have been applied to dissipative 2D MHD flows with periodic boundaries at Reynolds and Magnetic Reynolds numbers around the range of 100. Most recently, developments by Dellar [12] increased the tractability of these models by allowing for the independent control of the fluid viscosity and magnetic resistivity. All of these LBMHD results compared favorably to the conventional (e.g., pseudo-spectral) nonlinear macroscopic solvers. However, LB methods are not widely used in modeling problems of current interest to the plasma physics community [13].

There are three broad areas of plasma physics to which LBMHD should be applicable are: magnetic confinement fusion, magnetic reconnection and plasma astrophysics. All three areas will require new developments in LBMHD. Magnetic confinement fusion will require detailed treatment of toroidal geometry, realistic boundary conditions [4] and higher Reynolds and magnetic Reynolds numbers than have been achieved in current LBMHD codes. Additionally, non-local LBMHD closures capable of reproducing real kinetic effects will be needed for the solution of certain turbulent MHD phenomena. The problem of accurately predicting magnetic reconnection rates, although traditionally idealized to periodic 2D systems, also will require more sophisticated algorithms. In particular, the background flows orthogonal to the 2D plane require 3D lattices and the resulting thin filamentary current sheets require the use of adaptive grids for proper resolution of the fine scales. Furthermore, it has been shown [14] that resistive MHD alone cannot reproduce the correct reconnection rates. Finally problems in plasma astrophysics, like the magneto-rotational instability that is believed to drive accretion in black holes, will require the introduction of thermal effects into LBMHD.

In Section 3 we present some details on the scalar parallel and vector parallel computational performance of the LBMHD method on two high-performance computing platforms. In Section 4, we present some preliminary results on the inclusion in LB of critical kinetic effects that involve wave–particle interactions. Landau damping is a highly important and interesting phenomenon that was discovered in the linear theory of the Vlasov (collisionless Boltzmann) equation. It is understood as a wave–particle interaction in (continuous) kinetic space. LB methods are typically based on the simple linear BGK collision operator and have been used mostly to solve macroscopic nonlinear equations. Hence, at first glance, it would appear that Landau damping cannot be modeled by LB algorithms. Because of the need to incorporate Landau damping effects into complex plasma problems, effort was expended into developing macroscopic models that would incorporate the effects of Landau damping. A seminal paper was by Hammett and Perkins [15] who developed non-local kinetic closure schemes that recovered Landau damping on the macroscopic variables.

2. LBMHD formulation

In its simplest form, LBM is a linear discretized kinetic scheme solved on a minimal (kinetic) velocity lattice which reduces to fluid equations in the Chapman–Enskog limit. With α indexing the discrete set of velocity vectors $\boldsymbol{\xi}_\alpha$, LBM for Navier–Stokes takes the form

$$f_\alpha(\boldsymbol{x}+\boldsymbol{\xi}_\alpha\Delta t, t+\Delta t) = f_\alpha(x_i,t) - \frac{\Delta t}{\tau}[f_\alpha(\boldsymbol{x},t) - f_\alpha^{(eq)}(\boldsymbol{x},t)], \tag{1}$$

where the relaxation time for BGK collisions to drive $f_\alpha \to f_\alpha^{(eq)}$ is τ. One recovers the macroscopic fluid fields from the standard (discrete) moments of the scalar distribution function f_α.

The dissipative incompressible MHD equations are given by

$$\partial_t\rho + \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}) = 0, \tag{2}$$

$$\rho[\partial_t\boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}] = -\boldsymbol{\nabla}\left[P + \frac{B^2}{2}\right] + (\boldsymbol{B}\cdot\boldsymbol{\nabla})\boldsymbol{B} + \nu\rho\nabla^2\boldsymbol{u}, \tag{3}$$

$$\partial_t\boldsymbol{B} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{B} = (\boldsymbol{B}\cdot\boldsymbol{\nabla})\boldsymbol{u} + \boldsymbol{B}(\boldsymbol{\nabla}\cdot\boldsymbol{u}) + \mu\nabla^2\boldsymbol{B}. \tag{4}$$

In order to simulate MHD with LB, the second moment can be altered to include the non-linear terms originating from the Lorentz force or the force can be explicitly added to the LBGK equation [16]. Dellar [12] introduced a vector BGK kinetic equation for the evolution of the magnetic field $\boldsymbol{B}$

$$\boldsymbol{g}_\alpha(\boldsymbol{x}+\boldsymbol{\xi}_\alpha\Delta t, t+\Delta t) = \boldsymbol{g}_\alpha(x_i, t) - \frac{\Delta t}{\tau_m}[\boldsymbol{g}_\alpha(\boldsymbol{x}, t) - \boldsymbol{g}_\alpha^{(eq)}(\boldsymbol{x}, t)], \tag{5}$$

where τ_m is the magnetic relaxation time and one does not need to introduce a more complicated set of velocity streaming vectors. The magnetic field is recovered from a sum over this magnetic distribution function $\sum_\alpha \boldsymbol{g}_\alpha = \boldsymbol{B}(\boldsymbol{x}, t)$. While the Dellar model was for 2D MHD and used the d2q9m5 lattice, we have previously [17] applied Dellar's concept to the d2q9m9 octagonal lattice by decoupling the velocity space lattice from the spatial grid. LBMHD has recently been extended to 3D by Breyiannis and Valougeorgis [18] for the Hartmann flow. In general, the specific LBMHD distribution functions, which consist of a polynomial expansion in the macroscopic field variables, can be derived for a number of different lattice geometries in 2D and 3D. Details of the LBMHD scheme's application to Eqs. (2)–(4) in the incompressible and low Mach number limit are discussed in Ref. [12] along with the scheme's adherence to the $\nabla \cdot \boldsymbol{B} = 0$ condition.

3. Parallel performance of 2D LBMHD

We compare the performance of the 2D LBMHD code on two parallel computer systems, the IBM Power3 pSeries and the NEC Earth Simulator. These two systems are representative of a commodity processor-based system and a completely customized one. Table 1 shows some of the most pertinent characteristics of each platform.

3.1. Power3

The Power3 experiments reported here were conducted on the 380-node IBM pSeries system running AIX 5.1 and located at Lawrence Berkeley National Laboratory. Each 375 MHz processor contains two floating-point units (FPUs) that can issue a fused multiply-add (MADD) per cycle for a peak performance of 1.5 Gflops. The Power3 has a pipeline of only three cycles, thus using the registers more efficiently and diminishing the penalty for mis-predicted branches. The out-of-order architecture uses pre-fetching to reduce pipeline stalls due to cache misses. The CPU has a 32 KB instruction cache, a 128 KB 128-way set associative L1 data cache, and an 8 MB four-way set associative L2 cache with its own private bus. Each SMP node consists of 16 processors connected to main memory via a crossbar. The architecture is designed to recognize regular memory accesses and begin pre-fetching them to cache before they are required. SMP nodes are networked via the SP Switch2 (Colony) interconnect using a tree-like topology.

3.2. Earth Simulator

At the time of writing, the Earth Simulator (ES) is the world's most powerful supercomputer. The vector processor of the ES uses a dramatically different architectural approach than conventional cache-based systems. Vectorization exploits regularities in the computational structure of scientific applications to expedite

Table 1
Architectural highlights of the Power3 and ES computers

Pltfrm	Peak Gflops	MBW GB/s	Peak bytes/flop	MPI Lat. us	NBW GB/s/cpu
Power3	1.5	0.7	0.5	16.3	0.13
ES	8.0	32.0	4.0	5.6	1.5

The peak performance, memory bandwidth (MBW), number of bytes per floating point operation, MPI latency and network bandwidth (NBW) are shown.

uniform operations on independent data sets. The 500 MHz ES processor contains an 8-way replicated vector pipe capable of issuing a MADD each cycle, for a peak performance of 8.0 Gflops per CPU. The processors contain 72 vector registers, each holding 256 64-bit words (vector length of 256). For scalar instructions, the ES contains a 500 MHz superscalar processor with a 64 KB instruction cache, a 64 KB data cache, and 128 general-purpose registers. The 4-way superscalar unit has a peak of 1.0 Gflops ($\frac{1}{8}$ of the vector performance) and supports branch prediction, data pre-fetching, and out-of-order execution.

Like traditional vector architectures, the ES vector unit is a cache-less; memory latencies are masked by overlapping pipelined vector operations with memory fetches. The main memory chip for the ES uses a specially developed high-speed DRAM called FPLRAM (full pipelined RAM) operating at 24 ns bank cycle time. The ES contains 640 8-way SMP nodes connected through a custom single-stage crossbar. This high-bandwidth interconnect topology provides impressive communication characteristics, as all nodes are a single hop from one another. The 5120-processor ES runs Super-UX, a 64-bit Unix operating system based on System V-R3 with BSD4.2 communication features.

3.3. Structure of 2D LBMHD code

The 2D spatial grid is coupled to an octagonal streaming lattice and block distributed over a 2D processor grid. A set of mesoscopic variables is associated with each spatial grid-point, whose values are stored in vectors proportional to the number of streaming directions—in this case nine (eight plus the null vector). The simulation proceeds by a sequence of collision and stream steps.

A collision step involves data local only to that spatial point, allowing concurrent, dependence-free point updates; the mesoscopic variables at each point are updated through a complex algebraic expression originally derived from appropriate conservation laws. A stream step evolves the mesoscopic variables along the streaming lattice, necessitating communication between processors for grid points at the boundaries of the blocks. This is accomplished through MPI calls communicating between the four neighboring processors. Horizontal and vertical streaming vectors update only a single spatial grid-point, but diagonal vectors update three cells because a second-order interpolation is needed to connect the vector terminal to the spatial node. Overall, the stream operation requires interprocessor communication, dense and strided memory copies, as well as third-degree polynomial evaluation for the interpolation.

3.4. Porting details

Varying schemes were used in order to optimize the collision routine on each of the architectures. The basic computational structure consists of two nested loops over spatial grid points (typically 100–1000 loop iterations) with inner loops over velocity streaming vectors and magnetic field streaming vectors (typically 10–30 loop iterations), performing various algebraic expressions. For the Power3 system, the inner grid point loop was blocked to increase cache reuse leading to a modest improvement in performance for the largest grids and smallest concurrencies. For the ES, the inner grid point loop was taken inside the streaming loops and vectorized. The temporary arrays introduced were padded to reduce memory bank conflicts. No additional vectorization effort was required due to the data-parallel nature of the code.

3.5. Performance

Tables 2 and 3 present LBMHD performance on the studied architectures for grid sizes of 4096^2 and 8192^2. Note that to maximize performance the processor count is restricted to powers of two. The ES shows impressive results, achieving a speedup of approximately 44 × compared with the Power3 (for 64 processors). In fact, the 3.3 Tflops attained on 1024 processor of the ES represents the highest performance of LBMHD on any measured architecture to date. The efficiency of the code trails off somewhat going to higher processor counts.

There are two common metrics for gauging the efficiency of running on vector processors: the average vector length (AVL) which measures the average number of elements used per vector register, and should be close to the length of the vector registers for high efficiency; and the vector operation ratio (VOR) which

Table 2
LBMHD performance with a 4096^2 grid

Proc.	P3 Mflops/P	P3 % peak	ES Mflops/P	ES % peak
16	107	7	4616	58
64	142	9	4286	54
256	136	9	3211	40

Table 3
LBMHD performance with a 8192^2 grid

Proc.	P3 Mflops/P	P3 % peak	ES Mflops/P	ES % peak
64	105	7	4635	58
256	115	8	4259	53
1024	108	7	3297	41

measures the percentage of operations such as loads, stores, adds, etc. that are performed by the vector unit. The AVL and VOR are both near maximum for ES, indicating that this application is extremely well-suited for vectorization. Both metrics do decrease as the code is run at higher concurrencies as neither the MPI layer nor various program bookkeeping functions utilize the vector hardware to any large extent, and at higher concurrencies they become a more important fraction of the total time. The lower performance of the superscalar systems is first and foremost due to the less powerful processors. However, the relative performance as measured by percentage of peak is also considerably lower, mostly due to limited memory bandwidth. LBMHD has a relatively low computational intensity—about 1.5 FP operations per data word of access—making it extremely difficult for the memory sub-system to keep up with the arithmetic units. Vector systems are able to address this discrepancy through a superior memory system and support for deeply pipelined memory fetches. Additionally the 4096^2 and 8192^2 grids require 7.5 GB and 30 GB of memory, respectively, causing the sub-domain's memory footprint to exceed the cache size even at high concurrencies. Observe that superscalar performance relative to concurrency shows more complicated behavior than the vector systems. Since the cache-blocking algorithm for the collision step is not perfect, certain data distributions get better performance than others—accounting for increased performance at intermediate concurrencies. At larger concurrencies, the cost of communication begins to become more important, thus reducing performance despite better cache usage. This effect is also present in the ES results.

In the next two charts we show the percentage time spent in each of the main routines in LBMHD. The stream code section is further divided into the local part, stream, and the communication part labeled comm. Fig. 1 shows the results for the 8192^2 grid using 64 processors, and Fig. 2 shows the same simulation run on 256 processors.

Turning first to the ES results, the collision and stream steps perform with almost the same efficiency for 64 and 256 processors, but the communication step grows significantly. The message size has decreased from around 44 KB to 22 KB and the communication is becoming more and more latency bound. For the Power3 results, the efficiency of the collision step actually increases due to better cache use. The stream section has roughly constant performance and, similar to the case of the ES though not as pronounced, the cost of communication comes to depend more on switch latency.

4. Non-local closure for Landau damping

Landau damping is a great demonstration of irreversible macroscopic behavior in a collisionless (Vlasov) kinetic plasma due to wave–particle interactions. Because of this detailed kinetic interaction

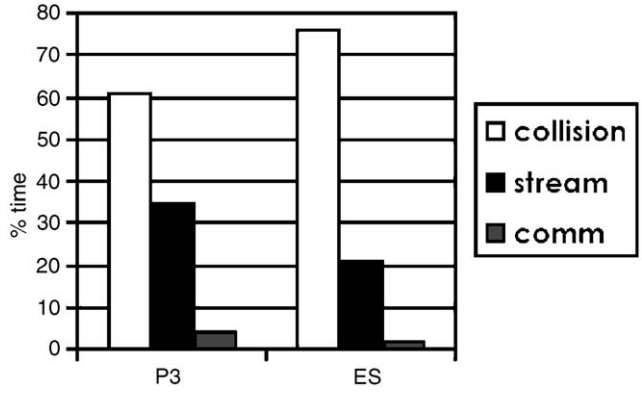

Fig. 1. Computational expenditure for a 8192^2 grid on 64 processors.

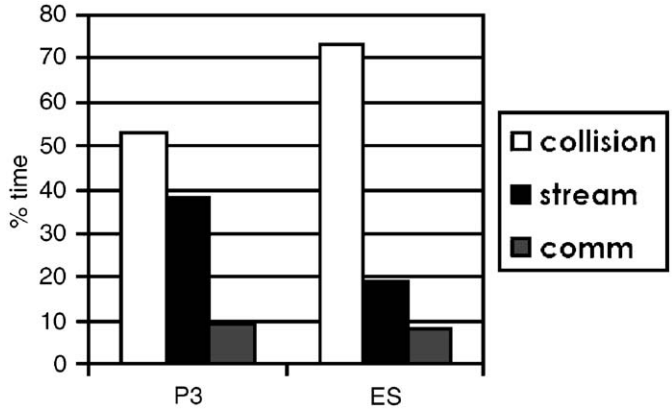

Fig. 2. Computational expenditure for a 8192^2 grid on 256 processors.

it had been thought for many years that Landau damping was completely outside the scope of (macroscopic) fluid equations. However, recent attempts at fluid descriptions of Landau damping have been successful [15,19] because of the introduction of non-local closures. The 1D electrostatic Vlasov equation is given by

$$\frac{\partial f}{\partial t}+\xi\frac{\partial f}{\partial x}+\frac{e}{m}E\frac{\partial f}{\partial \xi}=0, \tag{6}$$

where e/m is the charge-to-mass ratio and E is the electric field given by Poisson's equation. Defining the standard moments of density $\rho = m\int \mathrm{d}\xi f$, momentum $\rho u = m\int \mathrm{d}\xi \xi$, and pressure $p = m\int \mathrm{d}\xi f(\xi-u)^2$, we obtain the unclosed hierarchy

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x}(\rho u)=0, \tag{7}$$

$$\frac{\partial}{\partial t}(\rho u)+\frac{\partial}{\partial x}(\rho u^2)=\frac{\partial P}{\partial x}+\frac{e}{m}\rho E,\ldots\ . \tag{8}$$

The simplest model that exhibits the phase mixing of Landau damping is an appropriate closure ansatz at the zeroth moment and it will be this model that we will solve using a specific LB algorithm. The closure ansatz [15,19] at the level of the continuity equation is a diffusive Fick's law approximation

for the momentum

$$\rho u \approx -D\frac{\partial \rho}{\partial x}, \tag{9}$$

where D is a diffusion coefficient. The approximation for ρu can be written more explicitly in Fourier space

$$\Gamma_k = -D_k \mathrm{i}k\rho_k = -\sqrt{\frac{2}{\pi}}\frac{v_t}{|k|}(\mathrm{i}k\rho_k). \tag{10}$$

Transforming back into Cartesian space, the closure appears as a Hilbert transformation

$$\Gamma_x = \frac{\sqrt{2}v_t}{\pi^{3/2}}\int_0^\infty \mathrm{d}x' \frac{\rho(x+x') - \rho(x-x')}{x'}. \tag{11}$$

The continuity equation is thus written as

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x}\left[\frac{\sqrt{2}v_t}{\pi^{3/2}}\int_0^\infty \mathrm{d}x' \frac{\rho(x+x') - \rho(x-x')}{x'}\right]. \tag{12}$$

The LB method also requires a closure approximation for the highest moment variable that is introduced. This approximation is traditionally incorporated into the distribution functions themselves. For example, LB models for MHD and the Navier–Stokes equations traditionally approximate the pressure with the isothermal closure $P = \rho c_s^2$. We demonstrate here that non-local phase mixing closures can be incorporated into LB methods if they are added to the highest moment variable in the distribution functions. The price one pays for this level of sophistication, however, is that the evaluation of the collision step in the numerical algorithm is no longer a process that is completely local to the grid-point. It now depends on an integration over a significant portion of the neighboring space.

We present a simple demonstration of this methodology using the 1D phase mixed continuity equation described above. This simple model can be reproduced with a d1q3 LB formulation with the equilibrium distribution functions given by

$$f_\alpha^{(eq)} = \tfrac{1}{3}\rho + \tfrac{1}{2}\xi_\alpha \Gamma_x, \tag{13}$$

where $\xi_\alpha = (0, 1, -1)$. Note that the form of the integrand extends the integral over an infinite number of Riemann surfaces in the periodic domain. Thus for our purposes, the integral is truncated to only extend over the periodic domain, as the contributions from the exterior regions are negligible. The integral is taken with a second-order accurate integration scheme. Perturbations to the initial density profile will remain stagnant for the standard continuity equation but will decay away if phase mixing is correctly modeled. Fig. 3 shows the density profile for a 1D phase mixing LB simulation. The initial perturbation decays away as the system evolves. Fig. 4 shows the mean square of the density. The envelope clearly decays away due to the Landau

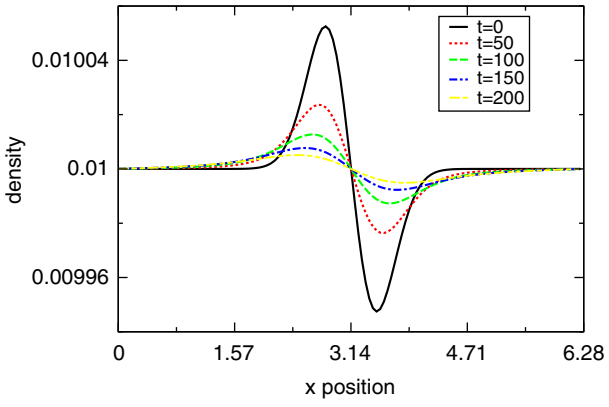

Fig. 3. The density profile for a 1D LB with phase mixing. The initial perturbation decays away in time.

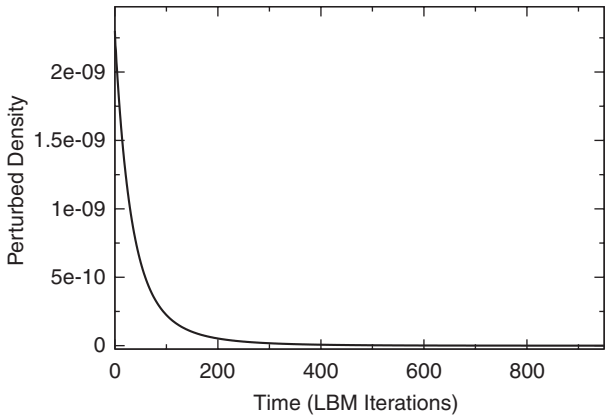

Fig. 4. The mean square of the density for a 1D LB with phase mixing. The envelope decays away due to kinetic phase mixing.

damping process. Direct comparison with Fig. 2 in Ref. [19] shows good agreement. We have thus demonstrated with this simple example, that LB methods are capable of incorporating fairly sophisticated techniques for reproducing some of the more subtle phenomena of plasma physics.

We are currently investigating non-local integral closures to the MHD equations such that the pressure is represented by

$$P = \frac{\sqrt{2}v_t}{\pi^{3/2}} \int_0^\infty \mathrm{d}x' \frac{1}{\boldsymbol{x}'} [\rho(\boldsymbol{x}+\boldsymbol{x}')\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{x}') - \rho(\boldsymbol{x}-\boldsymbol{x}')\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{x}')]. \tag{14}$$

And it would be desirable in the long term to treat a thermodynamic LBMHD model with similar non-local integral closure to the heat flux to achieve an even better approximation to the underlying kinetic physics.

5. Conclusion

We applied LBMHD methods to the simulation of some plasma physics problems. In particular, we have found that non-local kinetic closures can be incorporated into LB methods to reproduce the phenomena of phase mixing, and that LBMHD performs very efficiently in vector parallel computing environments. In the future, we plan to synthesize these efforts with other advancements to produce a concerted computational tool for the simulation of problems from magnetic confinement fusion, magnetic reconnection, and plasma astrophysics.

Acknowledgements

The authors would like to gratefully thank: the staff of the Earth Simulator Center, especially Dr. T. Sato, S. Kitawaki and Y. Tsuda, for their assistance during their visit; D. Parks and J. Snyder of NEC America for their help in porting applications to the ES. This research used the resources of NERSC at LBNL supported by the DOE under Contract no DE-AC03-76SF00098. This research was supported by the DOE Office of Science under Contracts numbered: (JC) DE-AC03-76SF00098, (GV,CS) DE-FG02-96ER54344 and (LV) DE-FG02-ER54360.

References

[1] S. Chen, H. Chen, D.O. Martinez, W.H. Matthaeus, Phys. Rev. Lett. 67 (1991) 3776.
[2] J.M. Koelman, Europhys. Lett. 15 (1991) 603.
[3] Y. Qian, D. d'Humieres, P. Lallemand, Europhys. Lett. 17 (1992) 479.
[4] S. Chen, D. Martinez, Phys. Fluids 8 (1996).

[5] L. Oliker, J. Carter, J. Shalf, D. Skinner, S. Ethier, R. Biswas, J. Djomehri, R. Van der Wijngaart, Concurrency Comput. J. Pract. Exper. 17 (2005) 69.
[6] S. Succi, The Lattice Boltzmann Equation, Oxford Press, Clarendon, 2001.
[7] D. Montgomery, G. Doolen, Phys. Lett. A 120 (1987) 229.
[8] D. Montgomery, G. Doolen, Complex Systems 1 (1987) 831.
[9] S. Succi, M. Vergassola, R. Benzi, Phys. Rev. A 43 (1991) 4521.
[10] S. Chen, D.O. Martinez, W.H. Matthaeus, H. Chen, J. Stat. Phys. 68 (1992) 533.
[11] D.O. Martinez, S. Chen, W.H. Matthaeus, Phys. Plasmas 1 (1994) 1850.
[12] P. Dellar, J. Comput. Phys. 179 (2002) 95.
[13] See for instance, the epitome of the 46th Annual Meeting of the Division of Plasma Physics of the American Physical Society, November 15–19, 2004.
[14] J. Birn, J.F. Drake, M.A. Shay, B.N. Rogers, J. Geophys. Res. 106 (2001) 3715.
[15] G.W. Hammett, F.W. Perkins, Phys. Rev. Lett. 64 (1990) 3019.
[16] H. Politano, A. Pouquet, P. Sulem, Phys. Plasmas 2 (1995) 2931.
[17] A. Macnab, G. Vahala, L. Vahala, P. Pavlo, M. Soe, Czech. J. Phys. 52 (2002) D59.
[18] G. Breyiannis, D. Valougeorgis, Phys. Rev. E 69 (2004) 065702 (R).
[19] G.W. Hammett, W. Dorland, F.W. Perkins, Phys. Fluids B 4 (1992) 2052.