[10] F. Bonomi and A. Kumar, *Adaptive optimal load balancing in a nonhomogeneous ... server system with a central job scheduler*, IEEE Trans. on Computers, v. 39, ... pp. 1232–1250, 1990

[11] J. Casas, R. Konoru, S.W. Otto, R. Prouty and J. Walpole, *Adaptive load mig... systems for PVM*, Proceeedings of Supercomputing '94, Washington DC, pp. 390... 1994

[12] M. Hamdi and C.K. Lee, *Dynamic load balancing of data parallel applications ... distributed network*, Proceedings of 1995 International Conference on S... computing, Barcelona, pp.170–179, 1995

[13] R. von Hanxleden and L.R. Scott, *Load balancing on message passing architec...* Journal of Parallel and Distributed Computing, v. 13, pp. 312–324, 1991

[14] R. Diekmann, B. Monien and R. Preis, *Load Balancing Strategies for Distributed Me... Machines*, Parallel and Distributed Processing for Computational Mechanics: Sy... and Tools, B.H.V. Topping (ed.), Saxe-Coburg, 1998

[15] T. Decker, M. Fischer, R. Lüling and S. Tschöke, *A Distributed Load Balanc... Algorithm for Heterogeneous Parallel Computing Systems*, Proceedings of the I... International Conference on Parallel and Distributed Processing Techniques ... Applications (PDPTA'98), H. R. Arabnia (ed.), CSREA Press, Volume II, pp. 93... 940, 1998

[16] http://www.esi.fr/products/crash/index.html

(69) / (70)

Iterative Momentum Relaxation for Fast Lattice-Boltzmann Simulations

D. Kandhai[1], A. Koponen[2], A. Hoekstra[1], and P.M.A. Sloot[1]

[1] Department of Mathematics, Computer Science, Physics and Astronomy,
University of Amsterdam,
Kruislaan 403 NL-1098 SJ Amsterdam,
Netherlands
email:{kandhai, alfons, sloot}@wins.uva.nl

[2] Department of Physics,
University of Jyväskylä,
P.O. Box 35, FIN-40351 Jyväskylä,
Finland
email:{antti.koponen}@phys.jyu.fi

Abstract. Lattice-Boltzmann simulations are often used for studying steady-state hydrodynamics. In these simulations, however, the complete time evolution starting from some initial condition is redundantly computed due to the transient nature of the scheme. In this article we present a refinement of body-force driven lattice-Boltzmann simulations that may reduce the simulation time significantly. This new technique is based on an iterative adjustment of the local body-force and is validated on a realistic test case, namely fluid flow in a static mixer reactor.

1 Introduction

The lattice-Boltzmann method (LBM) is a mesoscopic approach based on the kinetic Boltzmann equation for simulating fluid flow [1, 2, 3, 4]. In this method fluid is modeled by particles moving on a regular lattice. At each time step particles propagate to neighboring lattice points and re-distribute their velocities in a local collision phase. This inherent spatial and temporal locality of the update rules makes this method ideal for parallel computing [5]. During the last years, LBM has been successfully used for simulating many complex fluid-dynamical problems, such as suspension flows, multi-phase flows, and fluid flow in porous media. All these problems are quite difficult to simulate by conventional methods [3, 6, 7, 8].

However, as most numerical algorithms, the standard lattice-Boltzmann scheme also has its shortcomings. For instance, in a recently performed comparative study between the finite element and the lattice-Boltzmann method for simulating steady-state fluid flow in a SMRX static mixer reactor, it became evident that the computational time (on a sequential machine) required by the lattice-Boltzmann method was higher than that of the finite element method for obtaining the same level of accuracy. The memory requirements on the other hand

were lower for the lattice-Boltzmann simulations (details can be found in Ref. [9]). It can be argued that the longer computational time of LBM is a direct consequence of the transient nature of this scheme. In this article, we will present a new technique, namely the Iterative Momentum Relaxation technique (IMR), which can significantly reduce the saturation time. In this technique the body force which is often used to drive a flow in lattice-Boltzmann simulations, is adjusted dynamically by calculating the average loss of momentum due to viscous forces.

In section II we first review the basics of the lattice-Boltzmann method and the IMR technique. In section III we discuss a benchmark application, namely fluid flow in the SMRX reactor, and finally we present the results obtained with the IMR technique.

2 Simulation method

2.1 The lattice-Boltzmann BGK method

Basically, the time evolution of the lattice-Boltzmann model consists of a propagation phase, where particles move along lattice bonds from a lattice node to one of its neighbors, and a collision phase with a local redistribution of the particle densities subject to conservation of mass and momentum. The simplest and currently the most widely used lattice-Boltzmann model is the so-called lattice-BGK (Bhatnagar-Gross-Krook) model. Here the collision operator is based on a single-time relaxation to the local equilibrium distribution [2, 13].

The time evolution of the lattice-BGK model is given by [13]

$$f_i(\mathbf{r}+\mathbf{c}_i, t+1) = f_i(\mathbf{r},t) + \frac{1}{\tau}(f_i^{(0)}(\mathbf{r},t) - f_i(\mathbf{r},t)), \tag{1}$$

where $f_i(\mathbf{r},t)$ is the density of particles moving in the $\mathbf{c}_i$ direction, τ is the BGK relaxation parameter, $f_i^{(0)}(\mathbf{r},t)$ is the equilibrium distribution function towards which the particle population is relaxed. The hydrodynamic fields, such as the density ρ and the velocity $\mathbf{v}$, are obtained from moments of the discrete velocity distribution $f_i(\mathbf{r},t)$ (here N is the number of links per lattice point):

$$\rho(\mathbf{r},t) = \sum_{i=0}^{N} f_i(\mathbf{r},t) \qquad \text{and} \qquad \mathbf{v}(\mathbf{r},t) = \frac{\sum_{i=0}^{N} f_i(\mathbf{r},t)\mathbf{c}_i}{\rho(\mathbf{r},t)}, \tag{2}$$

and a common choice for the equilibrium distribution function is [13],

$$f_i^{(0)} = t_i\rho(1 + \frac{1}{c_s^2}(\mathbf{c}_i\cdot\mathbf{v}) + \frac{1}{2c_s^4}(\mathbf{c}_i\cdot\mathbf{v})^2 - \frac{1}{2c_s^2}v^2), \tag{3}$$

where t_i is a weight factor depending on the length of the vector $\mathbf{c}_i$, and c_s is the speed of sound. The lattice-Boltzmann model presented here yields the correct hydrodynamic behavior for an incompressible fluid in the limit of low Mach and Knudsen numbers [13].

Beside the computational kernel described above, flow simulations require a consistent set of boundary conditions for the solid walls and the in- and outlets. In Lattice-Boltzmann simulations solid walls are often imposed by using the bounce-back method, while inlet and outlets can be implemented by using pressure/velocity boundaries or body-forces[16, 17]. In the case of pressure/velocity boundaries the particle densities f_i at the inlet and outlet are chosen such that they yield some consistent values for the velocity or pressure. In the body-force approach, which is somewhat restricted to problems with a periodic geometry, the flow is driven by adding a fixed amount of momentum along the flow direction at each lattice point. The overall effect is that a pressure gradient is imposed between the inlet and outlet. For low Reynolds number flows, it has been shown for several benchmark problems that in the stationary state the hydrodynamic behavior of both the body-force and pressure/velocity boundaries are similar[17].

2.2 The Iterative Momentum Relaxation (IMR) Technique

As stated in our previous section, lattice-Boltzmann flow simulations are often driven by a body force. According to Newton's second law, the net force acting on the fluid phase during the simulation is equal to the rate of change of the total momentum,

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{Q} - \mathbf{T}(t), \tag{4}$$

where $\mathbf{P}(t)$ is the total momentum, $\mathbf{Q}$ is the total body-force and $\mathbf{T}(t)$ is the total viscous friction force due to the obstacles. In standard lattice-Boltzmann simulations the body-force is kept constant during the simulation, while the friction force depends on the velocity field and the geometry of the problem. A steady-state solution is reached when the total body force $\mathbf{Q}$ acting on the fluid is completely cancelled by the viscous friction force $\mathbf{T}$ due to the walls and obstacles.

The main idea of the IMR technique is to reduce the saturation time by adjusting the applied body force during the iteration depending on the change of fluid momentum at the iteration step considered. For some fixed amount of iteration steps (considered as a time interval in IMR) the momentum loss is computed and used to calculate the friction force acting on the fluid during that time interval as follows,

$$\mathbf{T}(t) = \mathbf{Q}(t) - \frac{d\mathbf{P}(t)}{dt}. \tag{5}$$

The body-force for the next time interval is then set equal to this guess. Notice that in this formulation, the body-force is no longer constant. Moreover, this strategy does not influence the explicit character of the Lattice-Boltzmann algorithm and thus its efficient and easy parallelization.

In summary, the IMR technique can be described by the following algorithm. First a flow is initialized. After every t_{step} time steps, the following iterative procedure (where k denotes the iteration counter of the IMR-loop) is repeated:

1. Calculate the momentum change $(\Delta P)_k$ of the fluid phase in the direction of the body force during the next time step.
2. Calculate the average momentum loss $T_k = Q_k - (\Delta P)_k$ (Q_k is the total body force at the iteration step k) of the fluid due to the viscous forces during this time step.
3. Choose a new body force as $Q_{k+1} = T_k$.

The new body force Q_{k+1} accelerates the fluid during t_{step} time steps before returning to step 1. The simulation is carried out until the body force Q and the total momentum reaches an acceptable degree of convergence. This is similar to the heuristical approach for the convergence criteria used in standard lattice-Boltzmann simulations.

3 Simulation results

To validate the IMR technique, we have simulated fluid flow in an SMRX static mixer reactor (see Ref. [9] for details). The SMRX static mixer reactor is a technology introduced 15 – 20 years ago, which has gained more and more in popularity within the chemical industry over recent years. It is a plug-flow type reactor filled with a series of SMRX static mixer elements (see Figure 1) turned at 90 degrees with respect to each other.

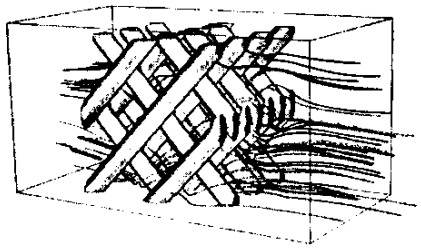
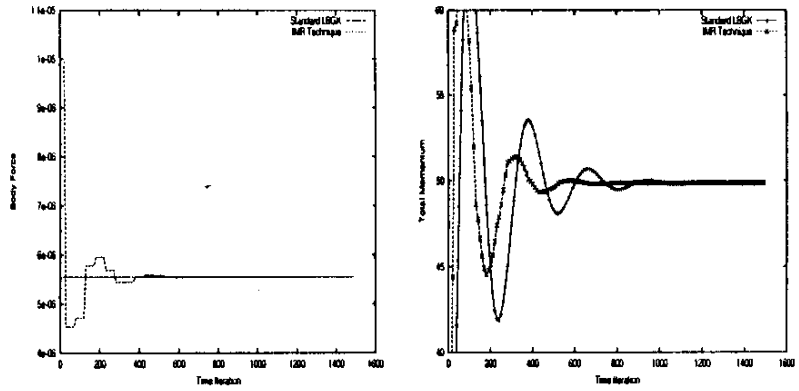


Fig. 1. The SMRX static mixer element. The reactor consists of an SMRX element placed in a rectangular duct. The inlet and outlet sections are of the same size as the element itself. The flow is from left to right. The streamlines illustrate the mixing process along the reactor.

The mixer element consists of specially designed stationary obstacles which promotes mixing of fluid flowing through it. Its mixing mechanism relies on splitting, stretching, reordering and recombination of the incoming fluid streams. In this communication we focus on only one SMRX element. Due to usually rather complex flows and geometries, only few 3D numerical simulations of flow through static mixers were performed in the past [15]. We have taken this application as a benchmark, since it is one of the very few cases of fluid flow in complex geometries with well documented results from traditional numerical methods and experimental data.



(a) (b)

Fig. 2. Left: Body-force as a function of the time iteration. Right: the time evolution of the total x-momentum along the SMRX reactor for the standard LBGK algorithm and the IMR technique is shown. $\tau = 1$ and the element dimensions are $56 \times 56 \times 56$ lattice points.

The time evolution of the body-force and the total momentum along the flow direction for the lattice-Boltzmann simulations with a constant body-force and the IMR technique is shown in Fig. 2. These simulations were performed for an element discretization of $56 \times 56 \times 56$ lattice points and the relaxation parameter τ was equal to 1. It is clear that the damping of the oscillatory behavior of the momentum is enhanced by the IMR technique. This is a result of the feedback of the flow field on the body-force. Moreover, both approaches clearly converge to the same value for the total momentum.

The time evolution of the relative difference in the total momentum along the flow direction, $\frac{|\Delta P_x|}{|P_x|}(t)$ (ΔP_x is computed between two results of two successive IMR trials), for the standard LBGK algorithm and the IMR technique is shown in Fig. 3. From this figure, it is evident that with the IMR method the relative difference converges faster to some level of tolerance.

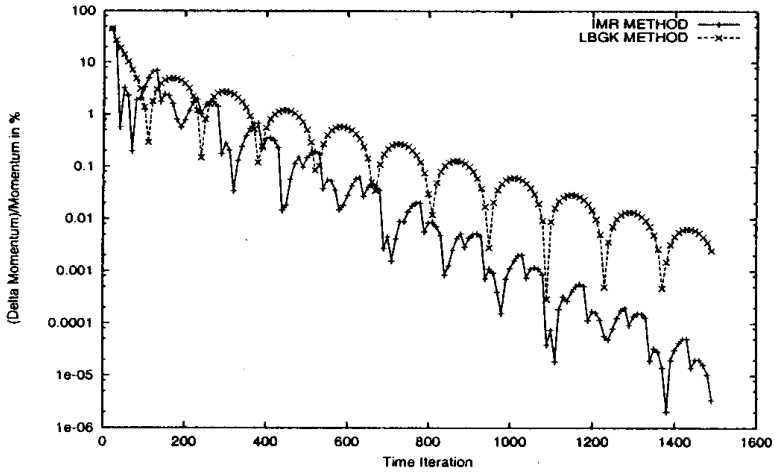
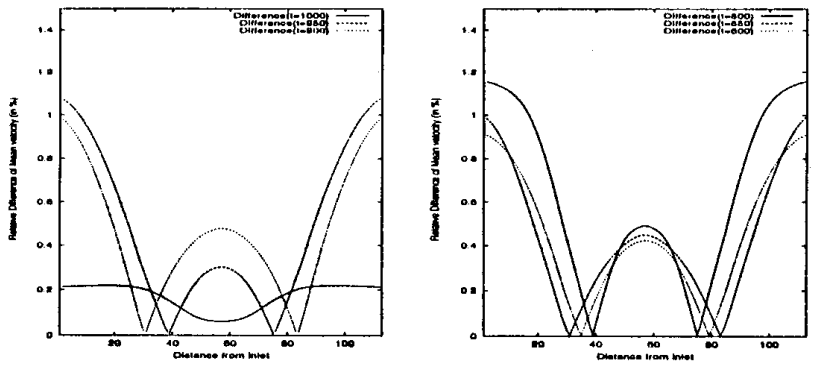


Fig. 3. The relative difference (in %) in the total momentum along the flow direction, $\frac{|\Delta P_x|}{|P_x|}(t)$, as a function of time, for the standard LBGK algorithm and the IMR technique. $\tau = 1$ and the element dimensions are $56 \times 56 \times 56$ lattice points. In both cases the oscillatory behavior is due to a non-zero initial velocity field. In the case of the IMR technique more oscillations are present due to the iterative refinement of the body-force.

In Fig. 4 we show the relative difference in the mean velocity along the reactor (in %) for different time-steps, in the case of the standard LBGK method (on the left) and the IMR technique (on the right). As reference data we have used the simulation results obtained after 1500 time-steps, as then the simulations were completely saturated in both cases. With the IMR technique 1% accuracy in the velocity and the pressure fields compared to the reference data, was already reached after 550 time steps, whereas the constant body-force method required around 1000 time steps to reach a similar accuracy. Moreover, the steady state solution of both approaches are very close to each other (data not shown). The relative difference in the mean velocity along the reactor, between the stationary state of both approaches, is smaller then 0.07%. In Ref. [9] we have shown in detail that the results of the standard LBGK method are also in good agreement with Finite Element calculations and experimental data. Thus we can conclude that the results obtained by the IMR technique are also consistent with experimental data.

In this test case we have used $t_{step} = 50$. Tests with some other values of t_{step} did not show significant improvements in the overall benefit gained by the IMR technique. Similar speed up results were also found for other Reynolds numbers provided that the flow is laminar. More detailed investigation and application of the IMR technique to other problems will be reported elsewhere[17].



(a) (b)

Fig. 4. On the left the relative difference in the mean velocity (in %) of the standard LBGK method is shown for $t = 900$, $t = 950$ and $t = 1000$. On the right the relative difference in the mean velocity (in %) of the IMR technique is shown for $t = 500$, $t = 550$ and $t = 600$. In both cases the relative difference is computed with respect to the simulation result at $t = 1500$ time-steps (simulation is then completely saturated) and the mean velocity is computed at different cross-sections along the reactor. $\tau = 1$ and the element dimensions are $56 \times 56 \times 56$ lattice points. The relative error is higher at the inlet and outlet, because the mean-velocity is smaller at those locations.

4 Conclusions

In many lattice-Boltzmann simulations, the complete time evolution of the system is computed with a constant body force starting from some initial velocity and pressure fields. The number of time steps which is required to reach the steady state can then be very large in some cases. We presented a new technique for reducing the number of time steps that is needed to reach the steady state for body-force driven flows. This strategy does not influence the explicit character of the Lattice-Boltzmann algorithm and thus its efficient and easy parallelization. We conclude that at least in problems involving laminar flow, the IMR technique can be very efficient in decreasing the number of time steps needed to reach the steady state.

Acknowledgments

This work was partly carried out within the MPR (Massive Parallel Computing) project "Many Particle Systems" funded by the Dutch foundation for basic