Mesoscopic Computational Haemodynamics

Artoli, A.M.M.

Citation for published version (APA):
Chapter 7

Optimisation Techniques

We present two adaptation techniques for the lattice Boltzmann method. First, the Mach number annealing is proposed for fast convergence of simulations of laminar time-dependent flows. The second technique is a test for the robustness of the method for interactive simulations. The first technique is an extension to the recent accelerated procedures for steady flow computations. Being based on Mach number annealing, the present technique substantially improves the accuracy and computational efficiency of the lattice Boltzmann method for such unsteady flows.

7.1 Necessity for Optimisation

In the previous chapters, we have demonstrated the suitability and investigated the accuracy of the standard lattice Boltzmann method in simulations of time-dependent fluid flows. We have also shown that use of curved boundary conditions significantly enhances the accuracy as compared to the bounce-back on the links. However, the bounce-back rule is still the most popular boundary condition, for its simple implementation and easy adaptation to complex geometry. Figure 7.1 shows temporal error behaviour in simulating a systolic cycle as a function of time, boundary conditions and Mach number. From this figure, it is shown that with the bounce-back, high order accuracy is only obtainable when the Mach number is very small (see the legend of the figure). This can be explained as follows. It is known that the bounce-back rule produces large errors of first order behaviour. In addition, simulations of time-dependent flows with the standard lattice Boltzmann model involve another major source of error: the compressibility errors. These two sources of error can be reduced significantly by reducing the Mach number. This, unfortunately, blows up the computational time needed for the simulation to converge. A current computational interest for all CFD solvers is to optimise simulation parameters for a desired accuracy with minimum computational cost. Within the lattice Boltzmann community, many efforts have been reported in this direction, mainly via implicit techniques (Tölke et al., 1998; Verberg and Ladd, 1999) local grid refinement (Filippova and Hänel, 1998; Yu et al., 2002) and scaling of the Reynolds number (Bernaschi et al., 2002). Most of these techniques
are applied to steady flows and/or affect the uniformity of the Cartesian grid and hence complicate parallelism in the computations. For unsteady flows, time evolution cannot be avoided and the method is computationally expensive, especially when the physical time scale is very small (which is a characteristic feature of dynamic complex systems).

In this study, we extend these acceleration techniques to unsteady flows. The idea is based on stepwise reduction of the Mach number after the simulation converges with a higher Mach number. We call this process Mach number annealing.

The standard lattice Boltzmann model works pretty well as long as the Mach number $M_a$ is low ($M_a^2 \ll 1$) and the density fluctuations are small. However, modelling unsteady flows involves higher density fluctuations, since the density and the unsteady pressure are tied up together through the ideal gas equation of state. Also, compressibility errors at high Mach numbers are expected. Although there exist a number of incompressible versions of lattice Boltzmann (e.g. He and Luo, 1997a; Guo et al., 2000) they have not been clearly formulated and tested in three dimensions, and are not yet highly thought of. A number of generalised lattice Boltzmann equations are recently gaining more attention (e.g. D'Humières, 1992). They provide more stable and accurate solutions, but at relatively higher computational cost. Here, we have applied the widely used quasi-incompressible D3Q19 model previously explained.
7.2 Mach Number Annealing

The Mach number is defined as the ratio between the speed $U$ of an object to the speed of sound

$$M_a = \frac{U}{c_s}.$$  (7.1)

Low-speed fluids ($M_a << 1$) can be considered as incompressible. As the Mach number approaches unity, compressibility effects need to be considered. The lattice BGK scheme involves a low-Mach number expansion of the Maxwell equilibrium distribution function and therefore, it introduces compressibility errors at relatively high Mach numbers.

In addition to the kinematic viscosity $v$, the diameter $D$ and the velocity $U$ which define the Reynolds number as $Re = \frac{UD}{v}$, a non-steady flow is characterised by a characteristic time interval, included in the Womersley parameter $\alpha = \frac{D}{2} \sqrt{\frac{\omega}{v}}$ or the Strouhal number, $St = \frac{Df}{U} = \frac{2\pi^2}{\nu k}$ where $\omega = 2\pi f = \frac{2\pi f}{k}$ is the angular frequency with $f$ being a typical frequency and $T$ the associated period of oscillation. An additional constraint comes from the fact that the accuracy of the lattice Boltzmann method reduces with increasing Mach number, especially for unsteady flows. The flow problem is completely defined by the geometry and these dimensionless numbers which take certain constant values. Now, in order to simulate at low-Mach number, we must decrease the velocity $U$ and consequently decrease the viscosity $v$ to produce the same Reynolds number. However, since the Womersley and the Strouhal numbers are dependent on the viscosity and the velocity, the frequency must also be reduced. Explained in formulae, the velocity $U$ is given by

$$U = \frac{Re}{D} = \frac{Df}{St} = M_a c_s$$  (7.2)

from which

$$M_a = \frac{Re}{c_s D}$$  (7.3)

and

$$Re = \frac{D^2 f}{v St}.$$  (7.4)

From these relations, we recognise that the Mach number $M_a$ and the kinematic viscosity $v$ are directly proportional to the frequency of oscillation through

$$M_a = \frac{fD}{St c_s}$$  (7.5)

$$v = \frac{fD^2}{St Re}$$  (7.6)

and

$$v = \frac{\pi D^2 f}{2\alpha^2}.$$  (7.7)
Eq. (7.5) implies that the frequency domain has to be reduced in order to have a low Mach number. This results in a considerable delay in the convergence behaviour. Eq. (7.6) shows that decreasing the frequency unfortunately results in pushing the simulation towards the instability region of the lattice BGK. Eq. (7.7) tells us that, for highly dynamic simulations (high $\alpha$), we need to consider both low frequency and viscosity. These constraints end up with a computationally expensive slowly time evolving simulation. This poses a high demand on a prospective acceleration method. An annealing process to accelerate the lattice Boltzmann method was first reported by Bernaschi et al. (2002). It allows fast convergence by combining viscosity annealing with powerful linear iterative solvers for computing the inverse Liouville operator. Different from those for steady flows, time-dependent lattice Boltzmann simulation parameters are not easy to control within a running simulation since, among others, new physical and hydrodynamic constraints need to be satisfied. The flow is now characterised by the Womersley number, the Reynolds number, and the Strouhal number, as discussed above. These parameters need to be fixed during annealing since the dynamics of the flow is highly time-dependent. We apply the same idea for unsteady flows, but anneal the Mach number instead of the Reynolds number on a strictly fixed spatial grid. We assume that the Mach number is to be annealed $n$ times and recall $n$ as the annealing factor. In order to do that

$$\frac{n}{M_a^i} = \frac{U}{U'} = \frac{\rho}{\rho'} = \frac{v}{v'} = \frac{7}{7} \quad (7.8)$$

which implies that all the velocity (in terms of the driving force), the frequency of oscillation and the viscosity are to be reduced $n$ times. This annealing strategy can be direct (1 level annealing) or multi-level. In the direct annealing strategy, after the simulation converges with a higher Mach number, the viscosity, the frequency and the driving force are reduced $n$ times in a single step and the simulation converges to the final solution. The multi-level annealing strategy involves gradual reduction of these parameters towards $n$, depending on the stability and tolerance constraints. In other words, there are different ways to decide when to start the annealing. Examples of both direct and multi-level annealing methods are discussed in the next section.

### 7.3 Simulations

We consider time-dependent systolic flow in a rigid tube of diameter $D = 63$ lattice units as a benchmark for our simulations. The first 8 harmonics of a pressure pulse, measured at the entrance of the human abdominal aorta, are used to apply an inlet condition for the tube. We have selected this complex time series for the sake of generality. For the outlets, constant density is applied. The velocity and the unknown distributions are computed from the density. For the walls, the bounce-back on the links is used. For all simulations the Womersley number is kept constant at $\alpha = 16$ and the average Reynolds number is $R_e = 270$. The simulation starts at average $M_a = 0.5 \quad (T = 360 \text{ and } v = 0.068)$ and waits until the system builds up its knowledge
7.3 Simulations

Figure 7.2: Obtained (dots) velocity profiles (left) and shear stress (middle) in lattice units during the systolic cycle (right), compared to the analytical Womersley solution (lines) for the 3D tube benchmark. The dots in the right column indicate times at which the profiles are shown. For this simulation $\alpha = 16$, $Re = 270$, and $Ma = 0.1$.

about the pulsatility and non-linear behaviour and converges after about 40 complete periods. Obtained simulation results are compared with the real part of the analytical Womersley solution given by Eq. (6.1).

The average error at $Ma = 0.5$ is 15%, originating from both compressibility effects and wall boundary conditions. Next, we reduce the Mach number to obtain good agreement with the analytical solution. We have previously studied the effect of reducing the Mach number on the accuracy for this benchmark (Artoli et al., 2002d). Fig. 7.2 shows sample simulation results for three different time frames after reducing the Mach number to $Ma = 0.1$. The new simulation parameters are computed from Eq. (7.8) after substituting $n = 5$ and including the initial simulation parameters. The average error is reduced to less than one per cent. However, since the period increases 5 times, the computational time increases with the same factor. The aim of Mach number annealing is to accelerate convergence to equilibrium by reducing the percentage tolerance in mass and momentum, computed by comparing similar points for each two successive periods. The mass tolerance is defined as

$$M \text{ tolerance } \% = \frac{M(t) - M(t - T)}{M(t - T)} \times 100$$

and the momentum tolerance is defined accordingly.

In typical simulations, we accept convergence below 0.1% for the momentum. We have performed three simulation sets for the systolic tube flow benchmark - one without annealing with the lowest desired Mach number, having $T = 1800$ and $v = 0.01353$. The pressure gradient is scaled to obtain a Mach number of 0.1. Fig. 7.3 (a, b) shows the relaxation of tolerance in mass and momentum from which we see that it takes
quite a long time to dampen the initial oscillations in tolerance (more than 72000 time-steps). The second set of simulations is conducted using 4 levels of annealing by reducing the Mach number after each 60 periods of the basic simulation. In detail, the Mach number is reduced to 0.4, 0.3, 0.2 and finally 0.1 directly after 60, 120, 180 and 240 complete periods of the basic simulation, respectively. The results are shown in Fig. 7.3 (c, d), from which we notice that the mass and more strongly the momentum converge much faster with the annealing process. The momentum tolerance is usually several orders of magnitudes higher than that for the mass, and hence, has more influence on the accuracy of the flow fields.

The third set shows a one-step annealing in which simulation parameters are directly tuned to the final Mach number \( M_a = 0.1 \) after convergence of the basic simulation in which \( T = 360 \) and \( M_a = 0.5 \). The direct annealing strategy significantly accelerates the relaxation towards equilibrium (see Fig. 7.3 (e,f)), since it significantly reduces compressibility errors earlier than the multi-level annealing process. For the non-annealed case, it takes a long time for the momentum to relax with a tolerance similar to the directly annealed simulations. In terms of numbers, the direct annealing strategy is at least 3 times faster for a 5 times annealed Mach number. The gain in computational time is higher if the ratio between the two Mach numbers is larger, since the order in the tolerance seems to depend only on the tolerance of the initial simulations rather than the annealing factor. The short-living spikes in Fig. 7.3 may be attributed to two reasons. First, since the systolic cycle is composed of many harmonic terms, values of point mass and momentum do not converge simultaneously. Compressibility errors at high velocities are also large. This explains why the spikes disappear with direct annealing, since the Mach number is reduced significantly.

### 7.4 Changing Geometry

In this section, we present the capability of the lattice Boltzmann method as a robust technique for interactive blood flow simulations, by considering the case of a phototypical symmetric bifurcation with a changing geometry. During vascular surgical planning, we envision that different geometrical solutions need to be tested on the patient's anatomical image provided by a suitable imaging technique. Conventionally, with NS solvers, for each newly suggested geometrical solution \( S_N \), the previous solution \( S_0 \) is discarded, a new grid \( G_N \) has to be generated and the simulation has to be restarted to obtain the solution \( S_N \). This may take considerable amount of simulation time. However, there are some CFD methods which are fully adaptive, such as the finite difference methods. In this section, we present primitive results on adaptivity of lattice Boltzmann method. Given a geometry \( G(t_0) \) at time \( t_0 \), we first run the lattice Boltzmann solver towards obtaining the solution \( S(t_0) \) while monitoring for a new geometry, not too different from the previous geometry. If at time \( t_i \) the user introduces a new geometry \( G(t_i) \), the simulation instantaneously adapts to the new grid and resumes towards obtaining the solution \( S(t_i) \) without a need to restart. The user may end up with a solution \( S_N \) for the geometry \( G(t_N) \). If the lattice Boltzmann
method is robust enough, the simulation time from $t_0$ to $t_N$ could be less than the sum of convergence times $T_i^0$ for each individual simulation, i.e. $t_N - t_0 < \sum_{i=0}^{N} T_i^0$. Moreover, the accuracy in $S_N$ must be the same as the solution $S_N(\text{restart})$ which is obtained by restarting the simulation. This is shown consequently.

We have conducted a number of 2D simulations on the bifurcation benchmark in-
Figure 7.4: A lattice Boltzmann Comparison between interactive simulations and restarted simulations in terms of simulation time for a symmetric bifurcation benchmark.

Introduced in the previous section, though allowing the bifurcation angle $\theta$ to change during the simulation after equal number of time steps. We have selected this benchmark for its similarity to the planning of a bypass for a diseased artery, where, the surgeon tries different paths to implant the host artery. The simulation starts at $t_0(\theta)$ at $\theta = 20^\circ$ and the system evolves towards the solution $S(0)$ a number of time steps $i$. At time $t = i$, the angle is increased by $\delta\theta$ and the simulation resumes towards the solution $S(i)$ for the geometry $G(i)$ another $i$ number of time step after which the geometry $G(2i)$ is introduced and so on, till we end up with $\theta = 80^\circ$ as our final $G(N)$ geometry. The simulation then converges to the solution $S(G_N(n))$. In lattice Boltzmann method, the system converges directly after the mass and momentum reach a given tolerance, chosen to be less than $10^{-5}$ for momentum and less than $10^{-9}$ for mass.

Technically speaking, the initialisation and the update of the new geometry are the critical factors which have direct influence on the total simulation time, while the choice of boundary conditions affects both stability and simulation time. In this experiment, we have tested two simple initialisation techniques. In both methods, only if the status of a node in the simulation box is changed from fluid to solid or from solid to fluid, will the node need initialisation. One way to initialise is to put these nodes to their equilibrium distributions which involves more computational time than the other simpler initialisation method such as assigning them to an average value. It is noted that the system forgets about the initialisation method in a short transitional time $t_{trans}$. Figure 7.4 shows the total number of nodes, the number of nodes to be updated and the total simulation time for interactive and restarted simulations. As shown in this figure, the total simulation time during an interactively changing geometry is in general smaller than the total simulation time for each individual sim-
ulation. This reflects a nice feature of the lattice Boltzmann method and makes it quite suitable for interactive simulation environments. However, the computational gain is less than 10% in this specific case. Application of acceleration techniques has proven to be feasible (Bernaschi et al., 2002). It is noted that other Cartesian grid CFD techniques may share this feature with the lattice Boltzmann method, but the body fitted grid solvers such as the finite element methods will be faced with the time it takes to adapt the new mesh. We are now using the lattice Boltzmann as a core simulation system for an interactive virtual vascular treatment environment using high level architecture (HLA) and a virtual 4D CAVE environment for interaction and visualisation (Belleman et al., 2000; Zhao et al., 2002).

7.5 Conclusions

In this chapter we have presented a numerical technique to accelerate laminar time-dependent lattice Boltzmann simulations through annealing of the Mach number during simulations, either directly or in a multi-level strategy. In both cases, the simulation is performed on a fixed grid and the viscosity, the Mach number, and the frequency are annealed by the same annealing factor. Considerable gain in computational time compared to that for the non-annealed standard lattice Boltzmann simulations is observed. We have shown that direct annealing of the Mach number is faster than the multi-level one. Since it works on the same grid, the Mach number annealing technique does not affect the parallelism of the uniform lattice Boltzmann Cartesian grid. Our current research concentrates on the optimisation of different annealing strategies of the Mach number for best acceleration. The capability of the lattice Boltzmann method for interactive simulations has shortly been demonstrated through studying flow in a changing geometry. It has been demonstrated that the lattice Boltzmann solver can be an adaptive flow solver without considerable difficulties.