

Benchmark computations based on Lattice-Boltzmann, Finite Element and Finite Volume Methods for laminar Flows

Sebastian Geller^{a,1} Manfred Krafczyk^a Jonas Tölke^a
Stefan Turek^b Jaroslav Hron^{b,1}

^a*Inst. for Computer Applications in Civil Eng., TU Braunschweig, Germany*

^b*Inst. for Applied Mathematics, LS III, University of Dortmund, Germany*

Abstract

The goal of this article is to contribute to a more precise discussion of the question whether Lattice-Boltzmann (LB) methods can be regarded as efficient CFD solvers. After a short review of the basic model and recommendable extensions, we compare the accuracy and computational efficiency of two research simulation codes based on the LB and the Finite-Element method (FEM) for incompressible laminar two-dimensional flow problems in complex geometries. As LB methods are weakly compressible by nature, we also study the influence of the Mach number on the solution by comparing compressible and incompressible results obtained by the LB code and the commercial code CFX. Our results indicate, that for the quantities studied (lift, drag, pressure drop) our LB prototype is at least competitive for incompressible transient problems, but asymptotically slower for steady-state Stokes flow as the asymptotic algorithmic complexity of the classical LB-method is not optimal compared to the multigrid solvers incorporated in the FEM and CFX code. For the weakly compressible case, the LB approach has a significant wall clock time advantage as compared to CFX. In addition, we demonstrate that the influence of the finite Mach number in LB simulations of incompressible flow is easily underestimated.

Key words: Lattice-Boltzmann; Finite Element; Finite Volume; CFD; benchmark;

¹ supported by the Deutsche Forschungsgemeinschaft DFG

1 Introduction

During the last decade much progress has been achieved in designing kinetic models of minimum complexity for fluid flow problems, especially in terms of the Lattice-Boltzmann (LB) method. For an introduction to the field we refer to [1–3] and the literature cited therein. Although it has been demonstrated, that LB methods allow to simulate a variety of complex flows, there is still insufficient evidence to demonstrate, that the computational efficiency of a LB-based CFD solver is comparable to a *state-of-the-art* solver based on the direct discretization of e.g. the incompressible Navier-Stokes equations. Recently there have been several works comparing the results of simulations based on kinetic and macroscopic methods [11–16], yet, either the details of the LB-models used there are unknown (as for the commercial PowerFlow code [17]) and any deviations in simulation results cannot be analysed or latest model extensions such as local grid refinement, improved second order boundary conditions, or Multiple Relaxation Time models (see below) have not yet been incorporated into the kinetic simulation prototype. Although most engineering fluid flow applications have to deal with the issue of turbulence, we restrict ourself in this work to the incompressible laminar flow regime. The paper is organized as follows: After a short reiteration of LB basics, we discuss their potential advantages and disadvantages for CFD simulations and motivate the necessity of some extensions utilized for the benchmark. We shortly sketch some properties of the Finite Element prototype and define the benchmark problems under consideration. The results of the computations together with the meshes used for the different methods are presented and discussed in the last two paragraphs.

2 Lattice-Boltzmann methods

2.1 Basics

The usual framework is to start from an evolution equation for particle distribution functions of the form

$$\vec{f}(\vec{r}_* + \vec{e}_i, t_* + \Delta t) - \vec{f}(\vec{r}_*, t_*) = \vec{\Omega}(\vec{f}(\vec{r}_*, t_*)), \quad i \in \{0, b\} \quad (1)$$

where exemplarily $\vec{f}(\vec{r}_*, t_*) = (f_0(\vec{r}_*, t_*), f_1(\vec{r}_*, t_*), \dots, f_b(\vec{r}_*, t_*))^T$ and b is the maximum number of discrete lattice vectors which generate the space-filling lattice with nodal unit distance Δx . In this work we use the D2Q9S model following the notation of [5], but we expect that our results are qualitatively

similar for other model variants in two and three dimensions with different sets of lattice vectors.

Although the set of equations (1) from a historic point of view has been derived from Lattice-Gas Automata, we prefer to regard it as a first order explicit upwind Finite Difference discretization of the discrete Boltzmann equation [6].

$$\frac{\partial \vec{f}}{\partial t} + \vec{c}_i \cdot \frac{\partial \vec{f}}{\partial \vec{x}} = \vec{\Omega}, \quad (2)$$

Here $\vec{c}_i = \frac{\vec{e}_i}{\Delta t}$. This interpretation helps to keep in mind that other discretizations of (2) than (1) are possible and have in fact been explored, e.g. [18–23]. We will shortly address this issue in the last section.

A typical choice of the quantities Ω_i in (1) is based on the Single Relaxation Time Approximation following [4]:

$$\Omega_i = -\frac{\Delta t}{\tau} (f_i(\vec{x}, t) - f_i^{(0)}(\vec{x}, t)) \quad (3)$$

where the equilibrium distribution functions are usually chosen as quadratic polynomials of the first two hydrodynamic moments [5], i.e. the density ρ and the momentum $\rho_0 \vec{u}$ which are related to the distributions by

$$\rho = \sum_0^b f_i \quad \rho_0 \vec{u} = \sum_0^b \vec{c}_i f_i. \quad (4)$$

In spite of the popularity of single relaxation time models it is well understood, that Multiple Relaxation Time (MRT) models have significant advantages in terms of stability and artefact minimization while qualitatively maintaining the algorithmic simplicity of the lattice Bhatnagar-Gross-Krook (LBGK) scheme [7–9]. The collision operator for the MRT model has the form

$$\vec{\Omega} = -\Delta t (\hat{M}^{-1} \hat{S} \hat{M} (\vec{f}(\vec{x}, t) - \vec{f}^{(0)}(\vec{x}, t))) \quad (5)$$

The matrix \hat{M} is used to transform \vec{f} into an equivalent vector $\vec{m} = \hat{M} \vec{f}$ in moment space (for a definition of \hat{M} see [9] and the relaxation matrix is given by $\hat{S} = \text{diag}(s_i), i \in \{0, b\}$). Choosing $\hat{S} = \text{diag}(\frac{1}{\tau})$ essentially reduces the MRT model to the BGK model, i.e. in case the relaxation times for the non-conserved moments are chosen identical. The basic collision algorithm of the MRT model for each grid node and timestep consists of

- transforming the nodal distributions into an equivalent set of moments ($\vec{m} = \hat{M}\vec{f}$),
- relaxing $b - 3$ (in 2D) non-conserved moments m_k with different relaxation times $\tau_k = \frac{1}{\omega_k}$ towards carefully chosen equilibrium moments $m_k^{(0)}$ (see [8])
- and retransform the relaxed moments back to f-space ($\vec{f}^{pr} = \hat{M}^{-1}\vec{m}^{pr}$) afterwards to obtain the distributions in a post relaxation (pr) state.

It can be shown by a multi-scale expansion [1], that the dynamics of the moments in (4) are solutions of the incompressible Navier-Stokes equation if it is ensured that the flow velocity is small compared to the sound speed of the LB system and the relaxation time (in units of Δt) is chosen as

$$\tau = \frac{\nu + \frac{1}{6} \frac{\Delta x^2}{\Delta t}}{c_s^2} \quad (6)$$

where ν is the kinematic viscosity of the Navier-Stokes momentum equation. For the D2Q9 MRT and LBGK models in this work the speed of sound is $c_s = \frac{1}{\sqrt{3}} \frac{\Delta x}{\Delta t}$. The pressure is related to the density by $p = c_s^2 \rho$.

A distinct advantage of the MRT model is the possibility to tune the non-hydrodynamic relaxation rates to minimize model artefacts and to optimize the numerical stability of the scheme. Yet, our simulations revealed a notable dependence of the results for drag and lift from these parameters. The mechanism of this influence is not well understood yet and requires further research. For the simulations described below, we used a value of 1.0 for the free relaxation rates (on the finest grid level) which gave the best results as was also found for some cases in [31].

2.2 Grid refinement

The use of locally refined meshes for the direct discretization of the Navier-Stokes equations is mandatory for an efficient solution of CFD problems. For Lattice-Boltzmann methods based on eq. (1), the problem of grid refinement is more complicated, as we do not start from a differential equation and the collision operator explicitly contains the numerical discretization parameters Δx and Δt . The usual way to implement grid refinement for LB methods is to keep the speed of sound constant on all grid levels. This leads to a nested time stepping scheme, i.e. for each grid level l we have $\Delta t_l = \Delta x_l$. The analysis of the LB equations on different grid levels reveals, that it is not sufficient to glue two cartesian grids of different solution together by interpolating the distributions during the propagation from the coarse to the fine grid. It was first shown in [24], that in order to obtain smooth transitions for pressure, velocity and stresses, one has to rescale the non-equilibrium distributions and

the nodal values for $\frac{\Delta t}{\tau}$ [19] to match the physical value of the kinematic viscosity. A clear description of details of the mesh refinement procedure can be found in [25,28,29,31].

2.3 Boundary conditions

In contrast to the case of a direct discretization of the Navier Stokes equation boundary conditions for macroscopic flow quantities can only be set implicitly via the boundary nodes' particle distribution functions. The simplest approach is to compute the equilibrium distributions from the desired values of the pressure and the velocity, but this is not sufficient for situations with non-zero stresses at the boundary. A well known and simple way to introduce no-slip walls is the so-called bounce back scheme which allows spatial second order accuracy if the boundary is aligned with one of the \vec{e}_i and first order otherwise. As the benchmark problem discussed in the subsequent sections has also curved boundaries, we utilize the boundary conditions developed in [32,33] for velocity Dirichlet boundaries. Thus we obtain second order accurate results in space even for curved geometries. In contrast to the simple bounce-back scheme the use of interpolation based no-slip boundary conditions results in a notable mass loss across the no-slip lines. Yet, the results obtained with bounce-back were inferior which highlights the importance of a proper geometric resolution of the flow domain. A constant pressure P_0 at the outlet is obtained by setting the incoming distributions to [35]

$$f_I = -f_i + f_I^{eq}(P_0, \vec{u}) + f_i^{eq}(P_0, \vec{u}) \quad (7)$$

where $\vec{e}_I = -\vec{e}_i$ and \vec{u} is obtained from eq. 4.

For a detailed discussion of LBE boundary conditions we refer to [34].

2.4 Implementation

The vast majority of LB implementations utilize matrices to store the nodal sets of distributions. This minimalist approach allows a very simple and efficient implementation in terms of floating point operations per second (FLOPS). If the volume of the bounding box of the computational domain is large compared to its actual flow volume, the use of full matrices is naturally inefficient in terms of memory consumption. Thus one can resort to use indirect addressing to spend grid nodes only where it is necessary for the computation. Although this requires substantial additional programming, one can achieve similar efficiency in terms of the nodal update rate per second (NUPS) as in the case of utilizing full matrices [26]. For topologically unstructured grids (in

the simplest case with uniform nodal distance but with holes in the computational domain) specifically tailored data structures can be developed [27]. In the general case, the optimal data structure is depending on the specific flow problem including the specified boundary conditions. In our approach, we utilize hierarchical grids (quadtrees in 2D and octrees in 3D) consisting of quadratic elements which can be recursively divided to obtain a local mesh refinement where the diameter of neighbouring elements may differ by a factor two. This approach allows an exponential refinement in space while keeping the programming effort limited. These data structures are also well suited for grid coarsening and thus allow also adaptive computations [28]. For our C++ research prototype *VirtualFluids* we obtain $\simeq 4 \times 10^6$ NUPS on a 2.0 GHz Opteron 64-bit processor for the D2Q9S model.

3 Direct discretization of the Navier-Stokes equation with Finite Element and Finite Volume techniques

3.1 FEM

FeatFlow (see www.featflow.de) is a (parallel) 2D and 3D FEM code for the solution of the incompressible nonstationary Navier-Stokes equations

$$\vec{u}_t - \nu \Delta \vec{u} + (\vec{u} \cdot \vec{\nabla}) \vec{u} + \vec{p} = \vec{f}, \quad \vec{\nabla} \cdot \vec{u} = 0 \quad (8)$$

which are discretized separately in space and time.

For the treatment of fully nonstationary flow configurations (via PP2D), the Navier-Stokes equations are discretized in time by one of the usual second order methods known from the treatment of ordinary differential equations (Fractional-Step- θ -scheme, Crank-Nicolson-scheme). Space discretization is performed by applying a special Finite Element approach using the non-conforming \tilde{Q}_1/Q_0 spaces (non-parametric version). The convective term is stabilized by applying an upwind scheme (weighted Samarskij upwind). Adaptive time stepping for this implicit approach is realized by estimating the local truncation error. Consequently, solutions at different time steps are compared. Within each time step the coupled problem is split into scalar subproblems using the *Discrete Projection method* [30]. One obtains definite problems in \vec{u} (Burgers equations) as well as in p (Pressure-Poisson problems). Then nonlinear problems in \vec{u} are treated with a fixed point defect correction method, the linearized nonsymmetric subproblems are solved by a multigrid approach. For the ill-conditioned linear problem in p , a special multigrid solver using Jacobi/SOR/ILU smoothers has been applied.

For low Reynolds numbers, i.e. steady flow configurations, a direct stationary approach (via CC2D) is used which treats the resulting nonlinear system after FEM discretization in space (with Q_2/P_1 elements and streamline-diffusion stabilization) with an outer Quasi-Newton solver while the resulting linear Oseen subproblems are solved via direct multigrid approaches (based on a "Vanka-like", resp., local pressure Schur complement smoother [30]). Numerical details regarding the FeatFlow methodology and the software implementation can be found at www.featflow.de and the references from this web site.

3.2 CFX

The commercial code CFX is a Finite Volume based CFD package with second order accuracy in space and time and an algebraic multigrid solver to provide optimal algorithmic complexity. CFX [36] is an integrated software system capable of solving diverse and complex three-dimensional fluid flow problems. The multi-grid based fluid flow solver provides solutions for incompressible or compressible, steady-state or transient, laminar or turbulent single- and multiphase and thermal fluid flow in complex geometries. The software uses unstructured and block-structured nonorthogonal grids with grid embedding and grid attaching to discretize the domain.

4 Benchmark definition

The flow geometry under consideration is depicted in Fig. 1. A channel of height H and width $4H$ is filled with cylindrical objects of diameter $H/8$ and distance $H/8$. The left vertical inflow line at $(x=0,y)$ as well as the top $(x,y=H)$ and bottom walls $(x,y=0)$ are subject to Dirichlet condition by imposing a velocity $U_0 = (u_x, 0)$ (including the right top and bottom corner nodes). The right vertical line $(x=4H,y)$ has a prescribed reference pressure P_0 .

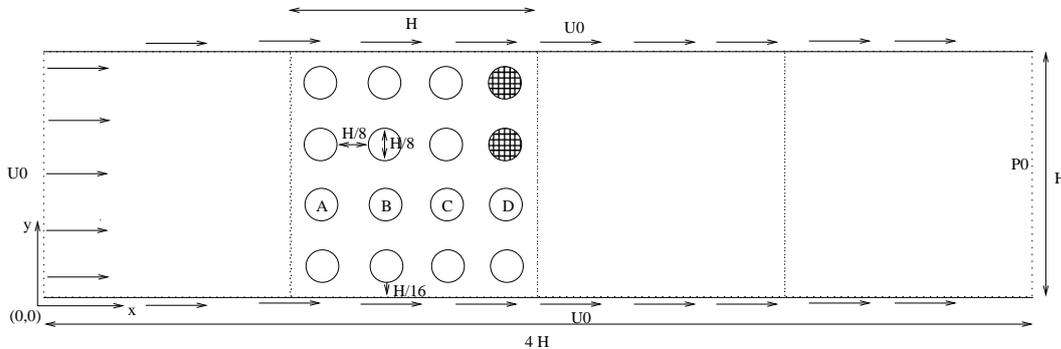


Fig. 1. Definition of the benchmark setup

We use the Ergun Reynolds number defined as

$$Re_E = \frac{U_0 D_P}{\nu(1 - \epsilon)} \quad (9)$$

where D_P is the particle diameter chosen as $H/8$, ν is the kinematic viscosity and ϵ is the porosity which we defined for simplicity as $\epsilon = 0.75$ for all cases.

In the following we simulated testcases with $Re_E = 1$ and $Re_E = 200$. For the $Re = 200$ case we omitted the two top right cylinders (checked in Fig. 1) to explicitly break the symmetry of the flow.

5 Results

A first test of consistency was a comparison of the pressure drop for $Re_E = 1$ (stationary case) across the system for the two different approaches. Choosing a roughly comparable number of degrees of freedom (DOF) (LB 549.936 DOF, FEM 297.858 DOF), Fig. 2 shows two pressure plots along two different horizontal lines ($y = 0.5/0.625H$). The results are virtually indistinguishable.

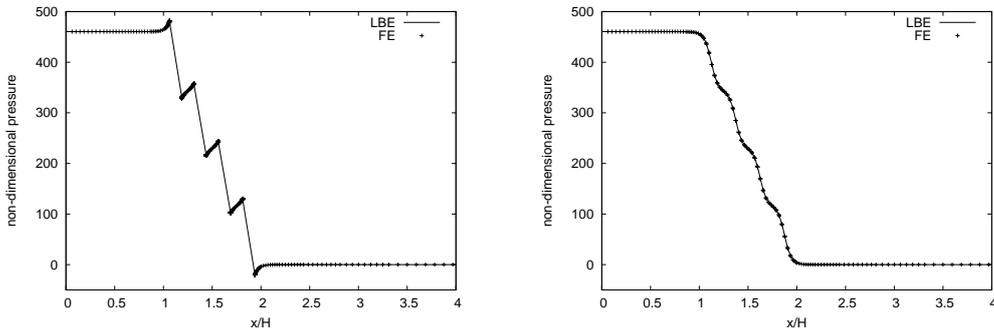


Fig. 2. Pressure plots for $Re_E = 1$ along horizontal lines ($y = 0.5/0.625H$)

As lift and drag are more sensitive quantities we compared the corresponding results for different meshes for both stationary and transient flows. Tables 1 and 2 show the corresponding values for different resolutions. All CPU-times were obtained on an AMD Opteron 64-bit 2.0 GHz system operating under Linux. Two typical meshes are depicted in Figs. 3 and 4. For these meshes the accuracy and efficiency is comparable as can be seen in Tables 1 and 2. The refinement labeling for the FE mesh is defined as follows: For the 1+2 case (see Fig.4) all initial elements are refined to level 1 (appr. a factor of 0.5 in Δx). Additionally, the base elements around (selected) cylinders are refined by another 2 levels. In the LB context a grid level p is equivalent to $\Delta x = H^{-p}$. The results for our grid convergence studies are given in Tables 3 and 4. For

the LB-solver the solution time grows quadratically with the number of DOF, whereas the FEM solver has a linear behaviour and is thus asymptotically more efficient.

Table 1

Drag: cylinder A, reference 465.58

| scheme | grid label | #dof | rel. error[%] | CPU-time[s] |
|--------|------------|---------|---------------|-------------|
| LB | 4-6 | 48.096 | 1.03 | 10 |
| LB | 4-7 | 141.696 | 0.43 | 93 |
| FEM | 0+2 | 11.774 | 0.77 | 6 |
| FEM | 1+2 | 26.922 | 0.58 | 53 |
| FEM | 2+0 | 30.642 | 1.34 | 33 |
| FEM | 2+1 | 43.314 | 0.37 | 38 |

Table 2

Lift: cylinder A, reference 0.9583

| scheme | grid label | #dof | rel. error[%] | CPU-time[s] |
|--------|------------|---------|---------------|-------------|
| LB | 4-6 | 48.096 | 1.07 | 10 |
| LB | 4-7 | 141.696 | 0.70 | 93 |
| FEM | 0+2 | 11.774 | 5.50 | 6 |
| FEM | 1+2 | 26.922 | 1.04 | 53 |
| FEM | 2+0 | 30.642 | 0.19 | 33 |
| FEM | 2+1 | 43.314 | 0.10 | 38 |

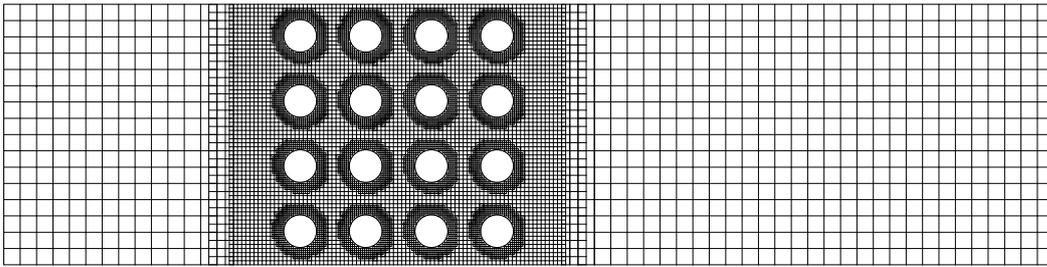


Fig. 3. Unstructured LB-grid (levels 4-7) with 141.696 DOF for $Re_E = 1$

The second test case is the transient flow problem at $Re_E = 200$ (cylinder D). We chose a non-symmetric flow geometry to make sure that the flow asymmetry does not depend on details of the mesh or the solver implementation. Figs. 6 and 7 show two typical grids which solve the problem for drag and lift to below 1% accuracy.

The reference solution was computed using a parallel CFX solver based on a very fine mesh. For the average drag we measured $c_d = 2.0548$, the average

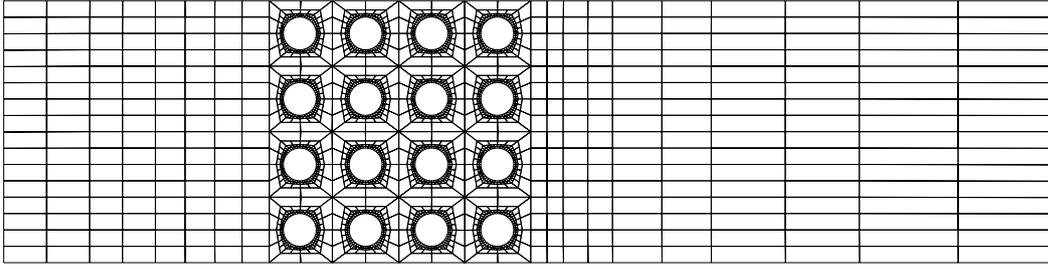


Fig. 4. Block-structured FEM-mesh labeled "1+2" with 26.922 DOF for $Re_E = 1$

Table 3

Drag convergence LB: circle A, $Re=1$, reference 465,58

| grid label | #dof | Mach number | drag | rel. error[%] | #timesteps | time[s] |
|------------|-----------|-------------|--------|---------------|------------|---------|
| 4-5 | 15.264 | 0.0173 | 450.76 | 3.183 | 3.900 | 3 |
| 4-6 | 40.320 | 0.0086 | 470.45 | 1.045 | 2.000 | 10 |
| 4-7 | 141.696 | 0.0043 | 467.57 | 0.427 | 2.900 | 93 |
| 4-8 | 549.936 | 0.0021 | 465.94 | 0.075 | 4.900 | 1.140 |
| 4-9 | 2.182.608 | 0.0010 | 465.65 | 0.014 | 7.900 | 14.700 |
| 9-9 | 8.976.960 | 0.0010 | 465.58 | 0.000 | 152.300 | 45.352 |

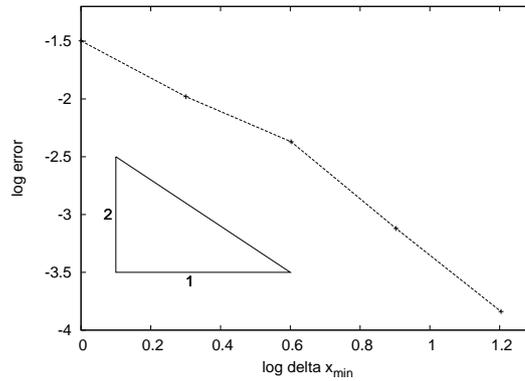


Fig. 5. LB: the spatial convergence for the Drag at $Re_E = 1$ is slightly better than second order

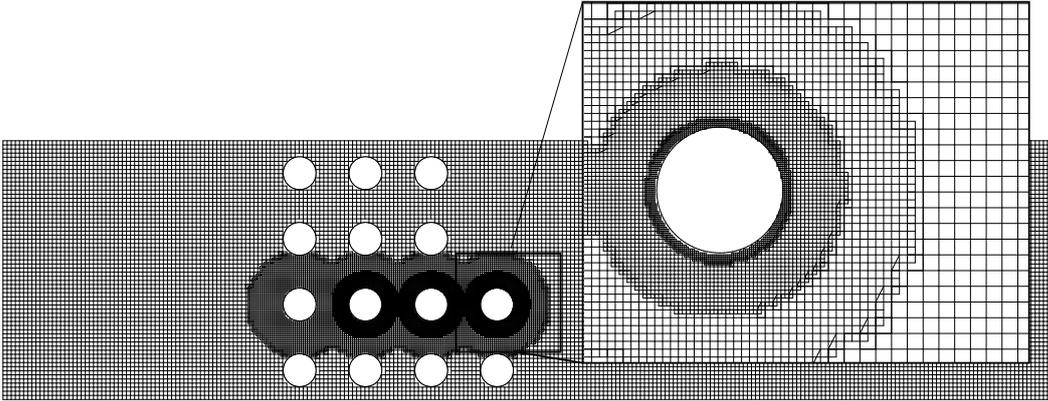
lift was $c_l = 0.9150$ and for the period of one drag oscillation we obtained $T_{ref} = 4.2327[\frac{D}{u_0}]$. Table 5 shows the results for the computed quantities for the three solvers. The Mach number for the LB-simulation was $Ma = 0.02$. For this simulation the LBGK approach based on a single relaxation time approach is unstable and the MRT approach is mandatory. Fig. 8 gives an impression of the transient lift and drag for the different methods used.

It should be noted that the CPU times for CFX could probably be reduced by choosing an optimized timestep size and more optimized meshes, but we do not expect a gain in computational efficiency of two orders of magnitude

Table 4

Drag convergence FEM: circle A, $Re=1$, reference 465,58

| method | mesh label | #dof | drag | rel.err[%] | time[s] |
|-----------|------------|-----------|--------|------------|---------|
| FEM-CC2D | (4) | 217.248 | 461.31 | 0.92 | 24 |
| FEM-CC2D | (5) | 864.576 | 464.51 | 0.23 | 121 |
| FEM-CC2D | (6) | 3.449.472 | 465.32 | 0.06 | 493 |
| FEM-Q2/P1 | (2+4) | 220.722 | 465.58 | 0.00 | 117 |
| FEM-Q2/P1 | (3+1) | 145.794 | 465.14 | 0.10 | 182 |
| FEM-Q2/P1 | (3+2) | 196.482 | 465.48 | 0.02 | 347 |
| FEM-Q2/P1 | (3+3) | 297.858 | 465.56 | 0.01 | 573 |
| FEM-Q2/P1 | (4+1) | 528.162 | 465.47 | 0.03 | 710 |
| FEM-Q2/P1 | (4+2) | 629.538 | 465.56 | 0.01 | 1.297 |
| FEM-Q2/P1 | (5+0) | 1.901.154 | 465.47 | 0.03 | 3.699 |
| FEM-Q2/P1 | (5+1) | 2.002.530 | 465.56 | 0.01 | 3.003 |
| FEM-Q2/P1 | (5+2) | 2.205.282 | 465.58 | 0.00 | 1.521 |

Fig. 6. Unstructured LB-grid (levels 6-9) with 243.774 DOF for $Re_E = 200$

from such measures. The times for both FEM and LBE should be interpreted in the sense that depending on local variations in the corresponding meshes or grids the required CPU time per period may vary notably, but will not change the trend implied by Table 5.

In the course of the simulations we found that the results of the LB-simulations showed a significant dependence from the chosen Mach number. In order to quantify this effect, we conducted a study where we computed flows with finite Mach numbers both with LBE and CFX. The results can be seen in Fig. 10. From these simulations it is obvious that the drag depends quadratically on the Mach number for both approaches as one would expect. For the tran-

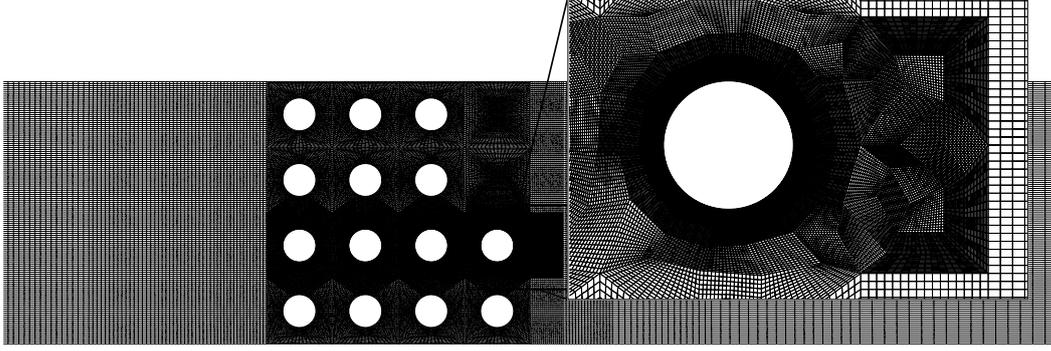


Fig. 7. Block-structured FEM-mesh labeled "(5)" with 450.528 DOF for $Re_E = 200$

Table 5

Results for the transient case (cylinder D):

columns 3-5 indicate relative errors of the corresponding quantities in %.

| scheme (grid label) | #dof | c_d [%] | c_l [%] | T_{ref} [%] | CPU-time[s] / T_{ref} |
|---------------------|-----------|-----------|-----------|---------------|-------------------------|
| LB (6-8) | 199.656 | 2.4 | 0.4 | 0.5 | 30 |
| LB (6-9) | 243.774 | 0.3 | 1.6 | 0.5 | 46 |
| FEM (4) | 113.264 | 1.3 | 0.3 | 9.2 | 22 |
| FEM (5) | 450.528 | 0.1 | 1.5 | 0.1 | 265 |
| CFX | 385.485 | 1.6 | 2.5 | 0.2 | 2.856 |
| CFX | 917.616 | 0.5 | 1.3 | 0.2 | 6.594 |
| CFX | 1.807.428 | 0 | 0 | 0 | 13.440 |

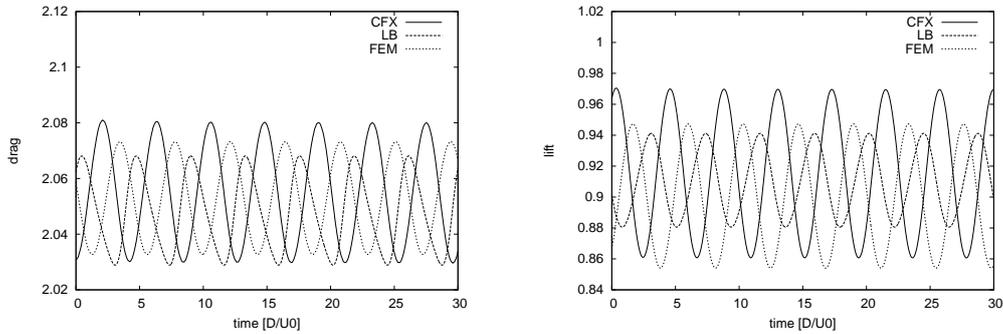


Fig. 8. Drag and Lift time series for LBE, FEM (high resolution) and CFX (reference) for $Re_E = 200$

sient case the difference between the drag results for the incompressible case and the case for $Ma = 0.1$ can easily exceed 10% for the average values as depicted in Fig. 10. In addition, higher modes are introduced. These typical finite Mach number effects should be kept in mind when using LB-methods

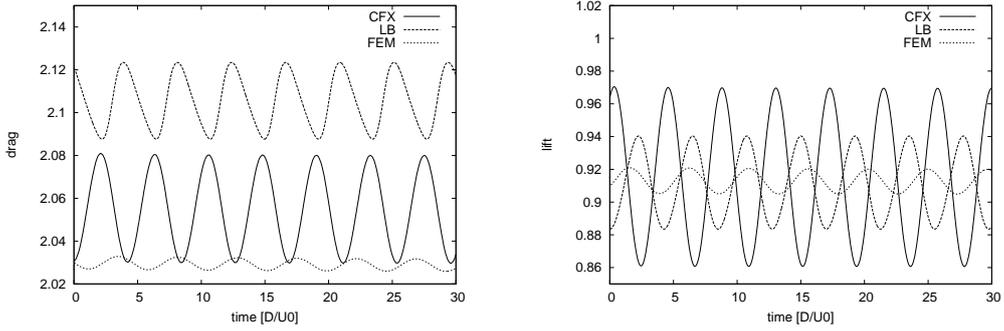


Fig. 9. Drag and Lift time series for coarser LBE, FEM (lower resolution) and CFX (reference) for $Re_E = 200$

for high-accuracy computations.

The required CPU-time for LB computations is approximately proportional to $\frac{1}{Ma}$ for a given finite Mach number flow problem with $Ma \leq 0.2$ (when using unstructured grids) whereas the CFX solution time is approximately constant as can be seen from Table 6 ². Thus one can expect a substantial advantage

Table 6

CPU times for the $Re = 200$ case drag computations with finite Mach numbers ($Ma = 0$ corresponds to the incompressible case).

| scheme (grid label) | #dof | $Ma[\%]$ | CPU-time[s] / T_{ref} |
|---------------------|---------|----------|-------------------------|
| LB (6-8) | 318.123 | 0.1 | 12 |
| LB (6-8) | 318.123 | 0.05 | 25 |
| LB (6-8) | 318.123 | 0.02 | 61 |
| CFX | 385.485 | 0.1 | 2.730 |
| CFX | 385.485 | 0.05 | 2.814 |
| CFX | 385.485 | 0.02 | 2.772 |
| CFX | 385.485 | 0 | 2.856 |

in computational efficiency when using LB-methods for flow problems where their built in physics in terms of finite Mach number is consciously utilized. The FEM solver used here is not applicable for compressible flows and thus cannot be used for the finite Mach number regime.

² The difference in DOF for LB (6-8) in comparison to Table 5 is due to different areas of refinement

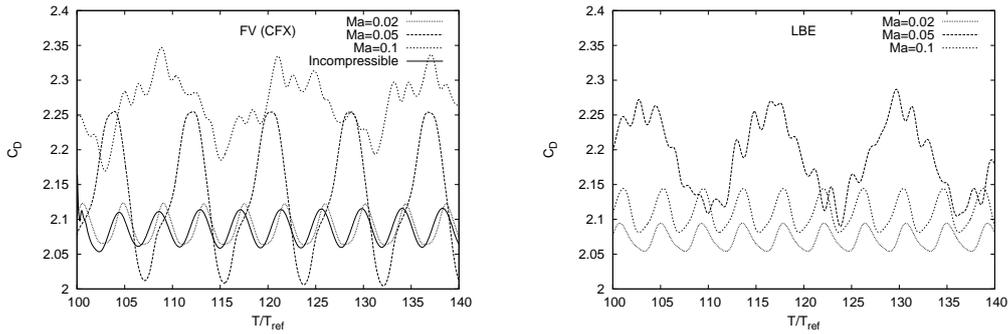


Fig. 10. Mach number dependance of drag results: Finite Volume (CFX) and LBE results

6 Discussion and outlook

From a theoretical point of view one would expect, that multi-grid based solvers have a principal advantage over the comparably simple numerical schemes utilized in the LB-approach, yet, at least for the transient test problems addressed in this work, these theoretical advantages did not manifest, as in the incompressible case the LB results while being of comparable accuracy were obtained at comparable or lower expenses in terms of CPU time compared to the FeatFlow version with fully adaptive time stepping and fully non-linear iteration as well as compared to CFX. For flows with small but finite Mach numbers, we note a substantial advantage in computational efficiency for the LB-approach in comparison with the Finite Volume-based CFX code. It will be interesting to see, if this advantage is persistent for the case when turbulence models are included.

For stationary laminar flows in complex geometries we found that the LB-approach is slower than the FEM approach which solves the stationary Navier-Stokes equation which in the end is no surprise as

- the basic LB algorithm of eq. (1) in its explicit form is asymptotically inferior in terms of computational complexity to a fully implicit approach for *stationary* problems
- and for small Reynolds numbers the type of the equation becomes more elliptic which increases the relative performance of the multigrid solver for the Poisson equation which has to be solved for the pressure.

Yet, these advantages can be transferred to the LB-approach, if one uses state-of-the-art approaches for solving the discrete Boltzmann equation as mentioned in section 2.1, but more research in this direction is required. The natural field of LBE based solvers are weakly compressible transient flows with Mach numbers up to around 0.2 because in this regime the Mach num-

ber influence on the time step size is moderate. Yet it was surprising to the authors that also for the stationary problems addressed in this benchmark, our LBE implementation was far more efficient than expected.

We hold the view that it is misleading to ask whether LB methods in general are better suited for CFD simulations than "conventional methods", because both terms are too general and one has to differentiate carefully between modelling errors (especially for complex fluids), discretization errors and implementational issues (which in the case of unstructured grids can have substantial influence on computational efficiency). A mandatory approach to conduct truly efficient simulations is the development of adaptive methods based on rigorous *a posteriori* error estimates. For the LB approach the authors are not aware of any related work except heuristic approaches [28,?]. Beside this issue, future work will address the issue of benchmarking turbulent flows in three dimensions as well as multiphase problems.

7 Acknowledgement

The financial support by the Deutsche Forschungsgemeinschaft in the framework of the *Lattice-Boltzmann Arbeitsgruppe* and the contributions of J. Bernsdorf, K. Beronov, G. Brenner, A. Klar, K. Steiner and T. Zeiser to the definition of the benchmark and valuable discussions are gratefully acknowledged.

References

- [1] D.A. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice-Boltzmann Models, Lecture Notes in Mathematics*, Springer Verlag, 2000.
- [2] S. Succi, *The Lattice Boltzmann equation, Oxford Science Publications* (2001).
- [3] D. Hänel, *Molekulare Gasdynamik* (in German), Springer Verlag, 2004.
- [4] P.L. Bhatnagar, E.P. Gross, M. Krook, *A Model for Collision Processes in Gases, Phys. Rev.* **94** (1954) 511.
- [5] Y. H. Qian, D. d'Humières, and P. Lallemand, *Lattice BGK models for Navier-Stokes equation, Europhys. Lett.* **17** (1992) 479-484.
- [6] X. He and L.-S. Luo, *Theory of lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, Physical Review E* **56(6)** (December 1997) 6811-6817.
- [7] D. d'Humières, *in Rarefied Gas Dynamics: Theory and Simulations, Prog. Astronaut. Aeronaut. Vol. 159* (1992), edited by B. D. Shizgal and D. P. Weaver AIAA, Washington, D.C.

- [8] P. Lallemand and L.-S. Luo, *Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability*, *Physical Review E* **61** (June 2000) 6546-6562.
- [9] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, *Multiple-relaxation-time lattice Boltzmann models in three-dimensions*, *Philosophical Transactions of Royal Society of London A* **360(1792)** (2002) 437-451.
- [10] I. Ginzburg, D. d’Humières, *Multi-reflection boundary conditions for Lattice-Boltzmann models*, *Phys. Rev. E* **68** (2003) 66614.
- [11] D. P. Lockard, L.-S. Luo, S. D. Milder, and B. A. Singer, *Evaluation of PowerFLOW for aerodynamic applications*, *Journal of Statistical Physics* **107(1/2)** (2002) 423-478.
- [12] M. Schäfer, S. Turek, *Benchmark computations of laminar flow over a cylinder*, *Notes in Numerical Fluid Mechanics* **52**, Vieweg Verlag Braunschweig (1996) 547-566.
- [13] D.R: Noble, J.G. Georgiadis, R.O. Buckius, *Comparison of Accuracy and Performance for Lattice Boltzmann and Finite Difference Simulations of Steady Viscous Flow*, *Int. J. Num. Meth. Fluids* **23** (1996) 1-18.
- [14] J. Bernsdorf, F. Durst, M. Schäfer, *Comparison of Cellular Automata and Finite Volume Techniques for simulation of incompressible Flow in Complex Geometries*, *Int. J. Num. Meth. Fluids* **29** (1999) 251-264.
- [15] D. Kandhai, D.J.-E. Vidal, A.G. Hoekstra, H. Hoefsloot, P. Iedema, P.M.A. Sloot, *Lattice-Boltzmann and Finite Element Simulation of Fluid Flow in a SMRX Static Mixer Reactor*, *Int. J. Num. Meth. Fluids* **31** (1999) 1019-1033.
- [16] Y.G. Lai, C.-L. Lin, J. Huang, *Accuracy and Efficiency Study of Lattice Boltzmann Method for Steady Flow Simulations*, *Numerical Heat Transfer Journal Part B: Fundamentals* **39** (2001) 21-43.
- [17] www.exa.com
- [18] J. Tölke, M. Krafczyk, E. Rank, R. Berrioz, *Discretization of the Boltzmann equation in velocity space using a Galerkin approach*, *Comp. Phys. Comm.* **129** (2000) 91-99.
- [19] J. Tölke, M. Krafczyk, Manuel Schulz, E. Rank, *Implicit discretization and non-uniform mesh refinement approaches for FD discretizations of LBGK Models*, *Int. J. of Mod. Phys. C* **9(8)** (1998) 1143-1157.
- [20] J. Tölke and M. Krafczyk and E. Rank. *A Multigrid-Solver for the Discrete Boltzmann-Equation*, *Journal of Statistical Physics* **107** (2002) 573-591.
- [21] T. Lee, C.-L. Lin, *A Characteristic Galerkin Method for Discrete Boltzmann Equation*, *J. Comp. Phys.* **171(1)** (2001) 336-356.
- [22] X. Shi, J. Lin, Z. Yu, *Discontinuous Galerkin spectral element lattice Boltzmann method on triangular element*, *Int. J. Num. Meth. Fluids* **42** (2003) 1249-1261.

- [23] F. Nanelli, S. Succi, *The lattice Boltzmann equation on irregular lattices*, *J. Stat. Phys.* **68(3/4)** (1992) 401.
- [24] O. Filippova and D. Hänel, *Boundary-Fitting and Local Grid Refinement for LBGK Models*, *Int. J. Mod. Phys. C* **(8)** (1998) 1271.
- [25] Dazhi Yu, Renwei Mei, and Wei Shyy, *A multi-block lattice Boltzmann method for viscous fluid flows*, *Int. J. Numer. Methods Fluids* **39(2)** (2002) 99-120.
- [26] M. Schulz, M. Krafczyk, J. Tölke and E. Rank, *Parallelization strategies and efficiency of CFD computations in complex geometries using Lattice Boltzmann methods on high-performance computers*, In: *High Performance Scientific and Engineering Computing*, Springer Verlag, 2001, 115-122.
- [27] S. Freudiger, *Effiziente Datenstrukturen für Lattice-Boltzmann-Simulationen in der computergestützten Strömungsmechanik*, *Diploma thesis*, TU München, 2001.
- [28] B. Crouse, E. Rank, M. Krafczyk, J. Tölke, *A LB-based approach for adaptive flow simulations*, *Int. J. of Modern Physics B* **17** (2002) 109-112.
- [29] B. Crouse, *Lattice-Boltzmann Strömungssimulationen auf Baumdatenstrukturen*, *Dissertation (in German)*, TU München, 2002.
- [30] S. Turek, *Efficient solvers for incompressible flow problems: An algorithmic and computational approach*, Springer (1999)
- [31] D. Yu, *Viscous Flow Computations with the Lattice Boltzmann equation method*, *PhD thesis*, Univ. of Florida, 2002.
- [32] M. Bouzidi, M. Firdaouss, P. Lallemand, *Momentum transfer of a Boltzmann-Lattice fluid with boundaries*, *Physics of Fluids* **13(11)** (2001) 3452-3459.
- [33] P. Lallemand, L.S. Luo, *Lattice Boltzmann method for moving boundaries*, *Journal of Computational Physics* **184** (2003) 406-421.
- [34] I. Ginzburg, D. d’Humières, *Multireflection boundary conditions for lattice Boltzmann models*, *Phys. Rev. E* **68** (2003) 066614-1-066614-30.
- [35] N. Thürey, *A single-phase free-surface Lattice-Boltzmann Method*, *Diploma thesis*, IMMD10, University of Erlangen-Nuremberg, 2003.
- [36] <http://www-waterloo.ansys.com/cfx/>
- [37] J. Tölke, S. Freudiger, M. Krafczyk, *An adaptive scheme for LBE Multiphase Flow simulations on hierarchical grids*, submitted to *Comp. & Fluids* 2004.