

Efficient monolithic simulation techniques for the stationary Lattice Boltzmann equation on general meshes

T. Hübner, S. Turek

Institute for Applied Mathematics, TU Dortmund, Germany

Abstract

In this paper¹, we present special discretization and solution techniques for the numerical simulation of the Lattice Boltzmann equation (LBE). In [11] the concept of the *generalized mean intensity* had been proposed for radiative transfer equations which we adapt here to the LBE, treating it as an analogous (semi-discretized) integro-differential equation with constant characteristics. Thus, we combine an efficient finite difference-like discretization based on short-characteristic upwinding techniques on unstructured, locally adapted grids with fast iterative solvers. The fully implicit treatment of the LBE leads to nonlinear systems which can be efficiently solved with the Newton method, even for a direct solution of the stationary LBE. With special exact preconditioning by the transport part due to the short-characteristic upwinding, we obtain an efficient linear solver for transport dominated configurations (macroscopic Stokes regime), while collision dominated cases (Navier-Stokes regime for larger Re numbers) are treated with a special block-diagonal preconditioning. Due to the new *generalized equilibrium formulation* (GEF) we can combine the advantages of both preconditioners, i.e. independence of the number of unknowns for convection-dominated cases with robustness for stiff configurations. We further improve the GEF approach by using hierarchical multigrid algorithms to obtain grid-independent convergence rates for a wide range of problem parameters, and provide representative results for various benchmark problems. Finally, we present quantitative comparisons between a highly optimized CFD-solver based on the Navier-Stokes equation (FeatFlow) and our new LBE solver (FeatLBE).

AMS Subject Classifications: 35A25, 65M06, 76D05, 76P05

Key words: Lattice Boltzmann equation, integro-differential equations, constant characteristics, upwind discretization, lower triangular matrices, Krylow-methods, multigrid, monolithic solver

1 Introduction

In this paper we consider the Lattice Boltzmann equation (LBE), sometimes also referred to as discrete velocity model (DVM) in contrast to the Lattice Boltzmann method (LBM) which is thoroughly discussed in the literature concerning stability [17], model extensions [4], boundary treatment [16], [32] or its connection to the Navier-Stokes equation [8], [21], [22]. Characteristic for the LBM is working on structured grids and mainly performing two explicitly decoupled substeps; first, particles stream along lattice vectors to the neighbouring nodes and, second, the particles are redistributed after collision. This method is unconditionally stable due to implicit discretization of the collision (see [9]), while the implicitness is removed using a modified distribution function in case of explicit on-lattice treatment of the advection. These ideas had been extended to off-lattice Boltzmann schemes as described in [1], but the authors explain that the remaining CFL restriction can be removed only by implicit treatment of the advection term. In fact, the DVM *is* a system of partial differential equations (PDE), and talking about modern Numerics for PDE one has to be aware of recent advances w.r.t. implicit time discretizations, unstructured, locally adapted grids and fast iterative solvers: They can provide more accurate results with less time steps and less grid points. A fully implicit, monolithic approach even allows direct stationary solvers in an efficient and robust way.

However, advection implicit Boltzmann schemes appear in only very few places. In [25] Tölke has applied an upwind finite difference discretization of second order on structured grids and used multigrid to solve the resulting linear equation systems. In [18] Mavriplis also took advantage of the multigrid method after

¹This work is kindly supported by the DFG through grant TU102/16-1 and the research center SFB/TRR 30 and partially by BMBF (SKALB project, grant 01IH08003D).

trying various iterative solvers for the resulting equation systems and encountering problems due to the high stiffness of the system and bad conditioning for realistic configurations. In [20] Noble introduced an implicit discretization of the LBM and successfully treated the resulting nonlinearities with the Newton method. For the resulting sparse banded matrices he used a solver from LAPACK and demonstrated for his scheme to be superior to the explicit method, with superlinear scaling of CPU time and memory demands of the Newton/LAPACK combination. It is obvious that the best direct linear solvers can perform only up to a certain memory limit, then, at the latest, iterative schemes have to take over. So, most of the mentioned authors applied on-lattice convection/advection implicit schemes. Here, we treat the LBE as a special (semi-discretized) integro-differential equation which consists of (linear) partial differential operators of transport-reaction type with constant characteristics. What is new is that we apply a special, higher order Finite Difference discretization on unstructured grids and combine it with implicit time discretization to achieve an efficient monolithic approach which works well especially for steady state problems. In the following, we consider the continuous Lattice Boltzmann equation to approximate the Navier-Stokes equation as in [21], namely

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = \frac{1}{\tau} (f_i^{eq} - f_i) \quad (1)$$

for the particle distribution function $f_i(\mathbf{x}, t)$ with single-time relaxation on a typical timescale τ and properly discretized phase space. In two dimensions, one set of discrete lattice vectors, that satisfies mass, momentum and energy conservation and has the needed symmetry (see [21]), is the D2Q9 model with the following lattice vectors:

$$\mathbf{e}_i \in \left\{ \begin{pmatrix} c \\ 0 \end{pmatrix}, \begin{pmatrix} c \\ c \end{pmatrix}, \begin{pmatrix} 0 \\ c \end{pmatrix}, \begin{pmatrix} -c \\ c \end{pmatrix}, \begin{pmatrix} -c \\ 0 \end{pmatrix}, \begin{pmatrix} -c \\ -c \end{pmatrix}, \begin{pmatrix} 0 \\ -c \end{pmatrix}, \begin{pmatrix} c \\ -c \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$$

The parameter c determines the speed of sound, $c_s = c/\sqrt{3}$, of the system, the viscosity ν is included via $\nu = c_s^2 \tau$ in the DVM (see [25]) and the distributions are summed up to get the moments of density and velocity:

$$\rho = \sum_i f_i \quad , \quad \rho_0 \cdot \mathbf{u} = \sum_i \mathbf{e}_i f_i$$

The Mach number $Ma = U/c_s$ is supposed to determine an additional error component of the system, namely the compressibility error of order $O(Ma^2)$ (see [7], [22]). The equilibrium term f^{eq} of the incompressible model (i.e. $\rho_0 = 1$, see [8]) introduces a nonlinear coupling of the distributions, if an implicit time-discretization is used:

$$f_i^{eq} = \omega_i \left(\rho + \rho_0 \left(\frac{3}{c^2} (\mathbf{e}_i \cdot \mathbf{u}) + \frac{9}{2c^4} (\mathbf{e}_i \cdot \mathbf{u})^2 - \frac{3}{2c^2} \mathbf{u}^2 \right) \right) \quad (2)$$

with

$$\omega_i = \begin{cases} 4/9 & , \quad \text{resting } \mathbf{e}_0 \\ 1/9 & , \quad \text{orthogonal } \mathbf{e}_i \\ 1/36 & , \quad \text{diagonal } \mathbf{e}_i \end{cases}$$

We want to emphasize that the parameter c appears as a linear scaling factor to the differential operator, and quadratic in the collision term through τ . This means that for small c and large viscosity ν the equation is transport dominated, while increasing c makes it collision dominated, the same for decreasing ν . We will discuss the interplay between c, h , and ν (see Sec. 2.4) which we discovered to be significant in the simulations since it vastly influences the approximation.

2 Discretization of the Lattice Boltzmann equation

Our approach to the discretization of the LBE will be covered by four main parts in this section. The first two parts represent special techniques from Numerics for PDE: Implicit time discretization will be described in Section 2.1; the second aspect is high order spatial discretization on unstructured grids which will be described in Section 2.2. The next part shortly describes the boundary treatment while in Section 2.4 we finally discuss the connection between spatial discretization and the choice of parameter c and how the combination of both affects the consistency of our model.

2.1 Implicit treatment of the generalized Lattice Boltzmann equation

To overcome typical stability restrictions, we want to treat the LBE in (1) fully implicitly to allow large time steps or even to solve directly for stationary flow, while a time-stepping scheme is needed only for configurations with high Reynolds number and for fully nonstationary flow problems. Therefore, we consider the following PDE describing the continuous Lattice Boltzmann equation in the generalized form

$$\alpha f_i + \mathbf{e}_i \cdot \nabla f_i = \frac{1}{\tau} (f_i^{eq} - f_i) + g_i \quad , i = 0, \dots, 8. \quad (3)$$

For the direct stationary treatment we set $\alpha = 0$ and $g_i = 0$, while we can also consider a nonstationary formulation by modification of the values, e.g.

$$\alpha = \frac{1}{\Delta t} \quad , \quad g_i = \frac{1}{\Delta t} f_i(t - \Delta t),$$

which corresponds to the implicit Euler time-stepping scheme. Here, we concentrate on the stationary case, $\alpha = 0$, while the first order implicit Euler approach and higher order schemes like Crank-Nicholson will be discussed along with nonstationary flow problems in a forthcoming paper [13]. For the stationary approach, the collision term on the right hand side of equation (3) becomes part of the system of algebraic equations to be solved. The collisions cause in every grid-point a coupling of the distributions in each direction, as part of a system matrix it means a 9x9 collision block each. Due to the nonlinearity in the equilibrium term, we have furthermore to apply a nonlinear solver in the stationary solution procedure. For this purpose we have to linearize the f_i^{eq} terms (in the way of $\mathbf{u}^2 \sim \mathbf{u} \cdot \mathbf{u}^{old}$) for calculating the Jacobian matrix in each Newton iteration step (see Section 3.1)

$$\begin{aligned} f_i^{eq} &= \omega_i \left(\sum_k f_k + 3 \sum_k D_{ik} f_k + \frac{9}{2c^2} \sum_k D_{ik} f_k (\mathbf{e}_i \cdot \mathbf{u}^{old}) - \frac{3}{2c} \left(\sum_k D_{1k} f_k u_1^{old} + \sum_k D_{2k} f_k u_2^{old} \right) \right) \\ &=: \sum_k \tilde{\omega}_{ik} f_k \end{aligned} \quad (4)$$

with $\tilde{\omega}_{ik} = \tilde{\omega}(i, k, \tau, c, \mathbf{u})$ and coefficients D_{ik} resulting from $(\mathbf{u} = (u_1, u_2)^T)$:

$$(\mathbf{e}_i \cdot \mathbf{u}) = c^2 \sum_k D_{ik} f_k \quad , \quad u_1 = c \sum_k D_{1k} f_k \quad , \quad u_2 = c \sum_k D_{2k} f_k$$

We summarize that due to the equilibrium term f_i^{eq} in the collisions, we have to solve a system of nonlinear, coupled algebraic equations including additionally a transport term which can be discretized using Finite Element/Difference/Volume methods. In our approach, we use finite difference upwinding on unstructured meshes which will be explained in the next section.

2.2 The "short-characteristic" discretization procedure

The transport operator in (3) is described in the standard LB method as a trivial streaming process of particles to neighbouring nodes, but since we allow general meshes, we will discretize it for each of the 8 constant characteristics using finite difference techniques. This task has been performed very efficiently in [10], [11] using a backward difference scheme of up to second order accuracy. In general the upwind discretization procedure works on unstructured triangular meshes as follows: With the unity vector \mathbf{n}_β for an arbitrary angle β (see Fig. 1), the constant characteristics upwinding of first order (to be denoted as upw1) yields the difference quotient

$$\mathbf{n}_\beta \cdot \nabla f(v_0) = f'(v_0) = \frac{f(v_0) - f(v_1)}{h_1} + O(h_1),$$

while the second order scheme (to be denoted as upw2) yields (with $h_1 + h_2 = r \cdot h_1$):

$$\mathbf{n}_\beta \cdot \nabla f(v_0) = f'(v_0) = \frac{-(1-r^2)f(v_0) - r^2f(v_1) + f(v_2)}{h_1(r^2 - r)} + O(h_1, h_2)^2$$

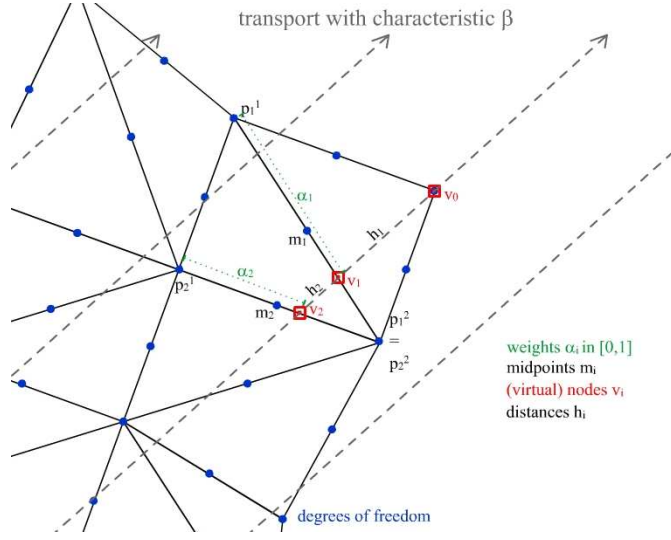


Figure 1: Setting for FD discretization of 1st and 2nd order

This means we discretize a one dimensional differential equation along the characteristic β . Analogously, the LB differential operator will be discretized (scaled by parameter c) as:

$$\mathbf{e}_i \cdot \nabla f_i = \begin{cases} \mathbf{c}\mathbf{n}_i \cdot \nabla f_i & , \text{ othogonal vectors} \\ \sqrt{2}\mathbf{c}\mathbf{n}_i \cdot \nabla f_i & , \text{ diagonal vectors} \end{cases}$$

Due to the upwind discretization using information from "backward" nodes, one is inclined to think that one can directly solve this transport equation starting from the inflow boundary and "traversing" the domain. In fact, as described in [11], for each constant characteristic, resp., lattice vector \mathbf{e}_i , it is possible to find a numbering of the grid nodes so that the resulting discretization matrix is lower triangular. This procedure has to be conducted once for each direction in preprocessing. Afterwards, whenever one has to solve a transport step (for example in preconditioning), it is possible without actually inverting a matrix but by simple backward insertion of the solution starting at the inflow boundary. Instead of describing the actual algorithm, which is based on topological sorting from the field of graph theory (see [2], [11]), we will visualize the result of the renumbering. After applying the according permutation matrices to the 8 transport parts situated on the block-diagonal of the system matrix, we see a change of the original allocation. Those entries, that belong to the finite difference discretization of the differential operator, are permuted into a lower triangular allocation in the matrix (Fig. 2). The off-diagonals contain the collision term. Additionally, exploiting the grid information we can implement this discretization in a matrix-free style, i.e. we can calculate all matrix entries including the finite difference and collision coefficients "on-the-fly".

2.3 Boundary treatment

The boundary treatment is in our case quite simple as we are using unstructured meshes. These are adapted to the geometry of our domain, therefore we can easily place our boundary nodes on the wall. To impose macroscopic quantities like velocity \mathbf{u}_{bc} as Dirichlet boundary conditions we use the scheme from [16]:

$$f_i = f_{-i} + 6\rho_0 \cdot \omega_i \frac{\mathbf{e}_i \cdot \mathbf{u}_{bc}}{c^2} \quad (5)$$

To achieve natural boundary conditions, i.e. a free-slip boundary, we use a different approach. It is similar to copying the distributions from a boundary layer before the free-slip nodes, as in the case of a structured lattice. In our case we apply constant extrapolation of the f_i , with $i = 1, \dots, 8$, from the domain in the direction being orthogonal to the wall. In the same way we proceed in corners where we have "pairs" of distributions f_i, f_{-i} both going into (or out of) the domain, in the case of a convex, resp., concave wall

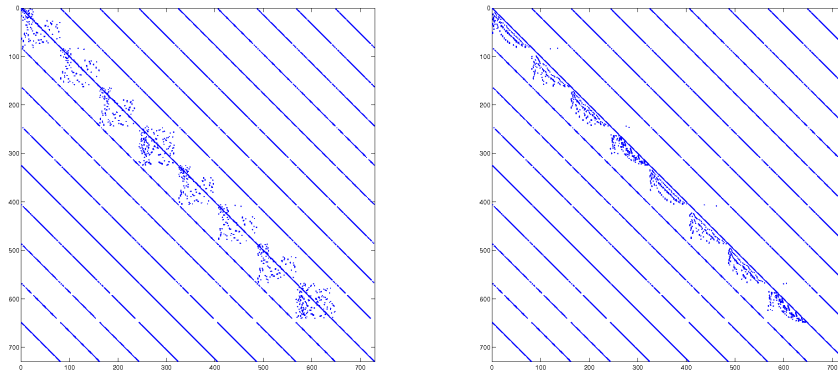


Figure 2: Matrix allocation: ordering by space index for each direction (left) vs. effect of renumbering algorithm in a symbolic representation (right)

(see [12]). For the stationary solution of the LBE we have to treat the boundary implicitly and include the dependence of the bounce-back distributions into the system matrix. This is necessary for the efficient treatment of the occurring nonlinearities as for example in the Newton scheme.

2.4 Convergence w.r.t. mesh size h and sound speed constant c

In common Lattice Boltzmann literature the parameter c is referred to as the unit of velocity, which couples grid length and timestep size through $\Delta x = c\Delta t$ (see [17]) and which is generally set to one. In our case we neither have a uniform grid spacing, nor a timestep when we solve the stationary LB equation on general meshes. Nevertheless we found that the choice of c influences significantly the accuracy and computational efficiency of our model. Starting the work on this topic it was not clear how to set the parameter c , which appears throughout our discretization. The idea to choose c as big as possible to reduce the Mach number proved to be wrong in the end. Looking at the equations from the viewpoint of asymptotical analysis for the LBE gave some hints that we must choose c depending on h (see [14], [15], [22]). Without the dependence of the two variables we would not achieve a consistent approximation of the Navier-Stokes equation in the asymptotic limit. Additionally, the order of our FD discretization of the differential operator is supposed to determine our choice, i.e. $h = O((1/c)^3)$ for first order upwind, resp., $h = O((1/c)^{3/2})$ for second order upwind. Only then we can achieve the desired quadratic convergence $O(\varepsilon^2)$ against the Navier-Stokes solution with $\varepsilon = \frac{1}{c}$. Corresponding numerical studies are presented in Section 5.2 and can be found in [12].

3 Solvers for the Lattice Boltzmann equation

3.1 Nonlinear solution method

Because of the terms $(\mathbf{e}_i \cdot \mathbf{u})^2$ and \mathbf{u}^2 in the collision term we have a system of nonlinear algebraic equations, denoted as

$$N(\mathbf{x})\mathbf{x} = g$$

with \mathbf{x} representing the solution vector for the distributions f_i . The nonlinearity can be solved by simple (damped) fixed-point iteration

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \omega N(\mathbf{x}^n)^{-1}(N(\mathbf{x}^n)\mathbf{x}^n - g) \quad , \quad \omega > 0$$

or - more efficiently - using the Newton method. To that purpose we write the system in residual form

$$R(\mathbf{x}) = N(\mathbf{x})\mathbf{x} - g = 0.$$

The Jacobian $\left[\frac{\partial R(\mathbf{x}^n)}{\partial \mathbf{x}}\right]$ is then used in the following iterative scheme:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \left[\frac{\partial R(\mathbf{x}^n)}{\partial \mathbf{x}}\right]^{-1} R(\mathbf{x}^n)$$

The Jacobian's coefficients in our model can be determined analytically: Summing up the partial derivatives of equation (4) yields:

$$df_i^{eq} = \omega_i \left(\sum_k f_k + 3 \sum_k D_{ik} f_k + \frac{9}{c^2} \sum_k D_{ik} f_k (\mathbf{e}_i \cdot \mathbf{u}^{old}) - \frac{3}{c} \left(\sum_k D_{1k} f_k u_1^{old} + \sum_k D_{2k} f_k u_2^{old} \right) \right)$$

So, the derivation simply results in a scaling by 2 in each quadratic term.

3.2 Iterative linear solvers

We need to solve a linear system $A\mathbf{x} = b$ (\mathbf{x} representing the discrete solution vector of all f_i) in each nonlinear step, having obtained a matrix A from the described discretization procedure. Direct solvers like LAPACK or UMFPACK are efficient for smaller problems, but scale badly with increasing number of unknowns, so that iterative methods have to be used for many gridpoints. Therefore, the simple (damped) Richardson iteration

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \sigma(A\mathbf{x}^{(n)} - b) \quad , \quad \sigma > 0$$

is a quite general candidate. This well-known, but usually very slow defect correction method, especially in the case of an ill-conditioned matrix A , can be accelerated by additional preconditioning

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \sigma C^{-1}(A\mathbf{x}^{(n)} - b)$$

where C should be easy to invert, and at the same time it should be similar to A so that $C^{-1}A \sim I$. Due to our special discretization we can obtain both properties using preconditioning by the transport part ("tr-pre") in the following. With our special numbering technique we can directly invert C consisting of lower triangular blocks and, in the case of a convection dominated system, we improve the condition number of $C^{-1}A$ significantly and independently of the mesh size. Another property of "tr-pre" is a strong clustering of the eigenvalues (see [12] for details), which is very favourable for Krylov-space methods like Bi-CGSTAB [30] and GMRES [23]. Unfortunately, for collision dominated configurations the positive effects diminish, therefore it makes sense to use the collision operator for preconditioning. Instead of numbering the unknowns according to the propagation direction, we "collect" in each grid point i its 9 unknowns f_0^i, \dots, f_8^i . Thus, we bring the offdiagonal matrix-entries onto the diagonal which then consists of a 9x9 block for each node. Consequently, a block-Jacobi ("bl-jac") preconditioner can be constructed by simply calculating the inverse of each 9x9 system. Unfortunately, the alternative numbering does not preserve the lower triangular form of the transport blocks. This preconditioner does not include the transport part and the condition of $C^{-1}A$ would therefore strongly depend on the mesh size. Finally, in the case of level-dependent convergence rates, an additional option is to apply the solver in a multigrid framework. To this purpose we need a set of hierarchical grids and transfer operations for prolongation and restriction which will be described in Section 5.5.

4 Generalized equilibrium formulation (GEF) of the LBE

In [11],[28] the concept of the *generalized mean intensity* (GMI) had been successfully combined with a direct transport solver to treat radiative transfer problems. The idea is to convey this technique from that integro-differential equation to the LBE: In this case we cannot improve the storage cost like in the GMI approach due to the varying coefficients $\tilde{\omega}_{ik}$. What we can do, though, is to combine the advantages of an efficient transport solver on the one hand with special preconditioning to deal with stiff problems due to dominant collision on the other hand.

In the following, we illustrate the procedure which is possible also for time-stepping variants of (3) (see also [13]). We start from the discretized LBE and denote by $\mathbf{T}_k f_k \sim \alpha f_k + \mathbf{e}_k \cdot \nabla f_k + \frac{1}{\tau} f_k$. Then, the equation

$$\mathbf{T}_k f_k = \frac{1}{\tau} f_k^{eq} + g_k \quad k = 0, \dots, 8 \quad (6)$$

can be formally rewritten as

$$f_k = \mathbf{T}_k^{-1} \left(\frac{1}{\tau} f_k^{eq} + g_k \right) \quad k = 0, \dots, 8 \quad (7)$$

and for each $i = 0, \dots, 8$ we multiply with the corresponding weights $\tilde{\omega}_{ik}$ from equation (4) which yields:

$$\tilde{\omega}_{ik} f_k = \tilde{\omega}_{ik} \mathbf{T}_k^{-1} \left(\frac{1}{\tau} f_k^{eq} + g_k \right) \quad k = 0, \dots, 8$$

Summing up over k finally gives us for each equilibrium term f_i^{eq} :

$$f_i^{eq} = \sum_k \tilde{\omega}_{ik} f_k = \sum_k \tilde{\omega}_{ik} \mathbf{T}_k^{-1} \frac{1}{\tau} f_k^{eq} + \sum_k \tilde{\omega}_{ik} \mathbf{T}_k^{-1} g_k \quad i = 0, \dots, 8$$

This means that in order to solve equation (6), we first solve for the terms f_i^{eq} and afterwards we obtain the f_i from a simple post-processing step according to equation (7). Thus, we have obtained the following new linear system in matrix form:

$$\left(\mathbf{I} - \begin{bmatrix} \tilde{\omega}_{00} \frac{1}{\tau} \mathbf{T}_0^{-1} & \tilde{\omega}_{01} \frac{1}{\tau} \mathbf{T}_1^{-1} & \dots & \tilde{\omega}_{08} \frac{1}{\tau} \mathbf{T}_8^{-1} \\ \tilde{\omega}_{10} \frac{1}{\tau} \mathbf{T}_0^{-1} & \tilde{\omega}_{11} \frac{1}{\tau} \mathbf{T}_1^{-1} & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\omega}_{80} \frac{1}{\tau} \mathbf{T}_0^{-1} & \dots & \dots & \tilde{\omega}_{88} \frac{1}{\tau} \mathbf{T}_8^{-1} \end{bmatrix} \right) \begin{bmatrix} f_0^{eq} \\ f_1^{eq} \\ \vdots \\ f_8^{eq} \end{bmatrix} = \begin{bmatrix} \sum_k \tilde{\omega}_{0k} \mathbf{T}_k^{-1} g_k \\ \sum_k \tilde{\omega}_{1k} \mathbf{T}_k^{-1} g_k \\ \vdots \\ \sum_k \tilde{\omega}_{8k} \mathbf{T}_k^{-1} g_k \end{bmatrix}$$

So, the resulting system matrix contains the transport steps in an inverse manner, which means that we obtain an *implicit* matrix only as we do not calculate the actual inverses T_i^{-1} , but we only apply the inverse to a vector which is equivalent to solving a linear system to a given right hand side. Due to our special discretization and the lower triangular form of the transport matrices T_i^{-1} , the whole procedure is very cost efficient. However, the new additional benefit of this GEF approach is that for the same reason we can apply additional preconditioning to the system, by collecting the entries in the 9x9 diagonals which are known since $diag(T_i^{-1}) = diag(T_i)^{-1}$ with T_i lower triangular. These diagonal entries consist in part of the $\tilde{\omega}(i, k, \tau, c, \mathbf{u})$, so we expect an additional stabilizing effect in the case of large c , while the solution of the GEF even without preconditioning should perform well for transport dominated configurations due to its explicit construction.

5 Numerical tests

5.1 Testcases and hierarchical grids

As testcases we choose three different "classical" problems from the field of CFD with varying difficulty level. First, we compare with the given analytical solution for *Rotating Couette flow* from [6]. Then, additional simulations are performed for the *Flow around Cylinder* benchmark (see [24]), resp., for the *Driven Cavity* problem at various Reynolds numbers (as in [18]). Due to the lack of analytical solutions, we compare against Navier-Stokes reference values computed by FeatFlow [27] on a highly refined mesh.

As seen in Fig. 1, our spatial discretization is designed for general (triangular) grids. Thereby, we have 3 degrees of freedom (DOF) per element for the 1st order upwinding, and for the 2nd order upwinding we have 3 additional unknowns in the edge midpoints. Starting from a coarse mesh we get one further level of refinement - and successively a mesh hierarchy used in a multigrid algorithm - simply by connecting the edge midpoints. Thus, we divide each element into 4 new ones. In Fig. 3 we present a structured grid on the unit square, used for the *Driven Cavity* testcase, together with another less structured grid which we use for the *Rotating Couette flow*. Furthermore, in Fig. 4 a sequence of grids is shown which is locally adapted around the inner boundary component. On these grids we calculate the *Flow around Cylinder* benchmark.

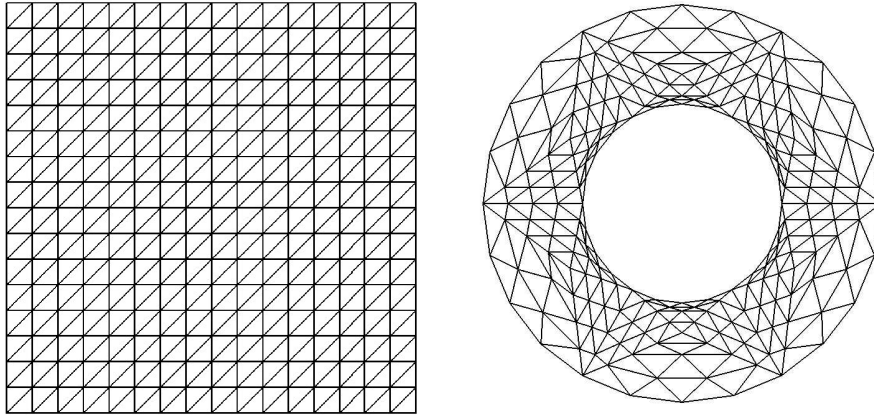


Figure 3: Structured and unstructured grid for *Driven Cavity* and *Rotating Couette flow*

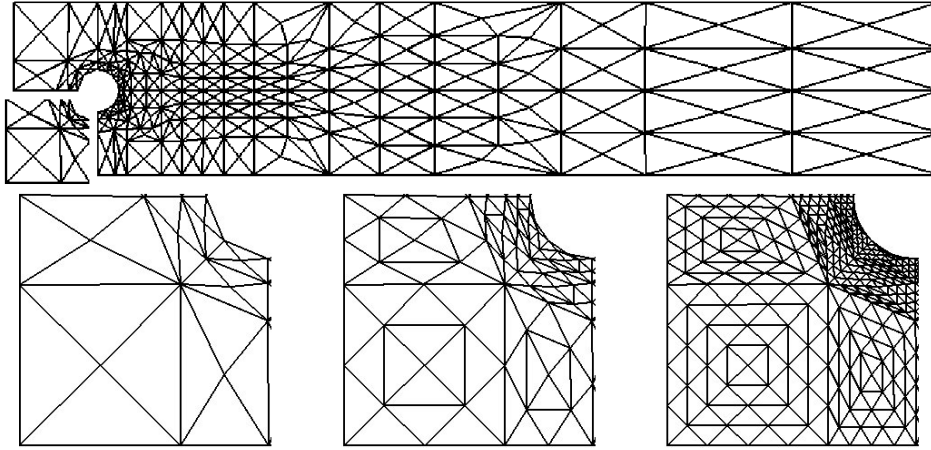


Figure 4: Hierarchical sequence of refined, locally adapted grids for *Flow around Cylinder*

To better approximate the forces acting on the circular cylinder, we have more elements gathered around the cylinder than at the channel's outflow region. Nevertheless, the upwind discretization is carried out similarly on all three meshes.

5.2 Results for the spatial accuracy

To verify the accuracy of our discretization techniques (see Section 2.4) we compute the relative L_2 -error of velocity \mathbf{u} for all three testcases, using either the given analytical or CFD reference solution. We choose an initial grid width h_0 and sound constant c_0 . During grid refinement we fix the ratio of both according to the order of discretization, i.e. $h_l \cdot (c_l)^3 = h_0 \cdot c_0$ (upw1), resp., $h_l \cdot (c_l)^{3/2} = h_0 \cdot c_0$ (upw2). In Fig. 5 we plot the theoretical slope of quadratic convergence, the points in the slope indicating the distance between successive levels of refinement. The results show a good accordance with the theory, neglecting the initial results as we are interested mainly in the asymptotic behaviour. Compared to the 1st order upwinding, the 2nd order discretization obviously reduces the error significantly with less grid refinement steps, at the cost of bigger choice of c .

Although the asymptotic behaviour is clear, it is not obvious which initial c_0 to choose. That is why we continue with the numerical analysis, now varying c for each fixed h_l . The plots show (see Figs. 6, 7) the existence of a (unique) optimal choice, say a c_{opt} , for each testcase and refinement level. Using a finer mesh also increases the $c_{opt} = c_{opt}(h)$. Furthermore, c_{opt} is getting smaller for increasing Re (see Fig. 8), so it shifts together with the viscosity ν of the simulated medium. This faces us two problems, one is that for both, increasing Re and smaller c , the nonlinearity is more difficult to solve. The other is the increased compressibility error for larger Mach numbers. As remedy it is common to use multiple-relaxation-time models (MRT, see [3], [4], [5], [17]) which are subject of our current research and which will also be included in a forthcoming paper. So far we could derive a rule of thumb for the choice of c in each testcase, which is also relevant for the linear and nonlinear solvers. We should keep in mind that small c means transport dominated equations, while big c means dominant collisions. On the other hand, c_{opt} seems to be in an intermediate range, where neither effect really dominates the other.

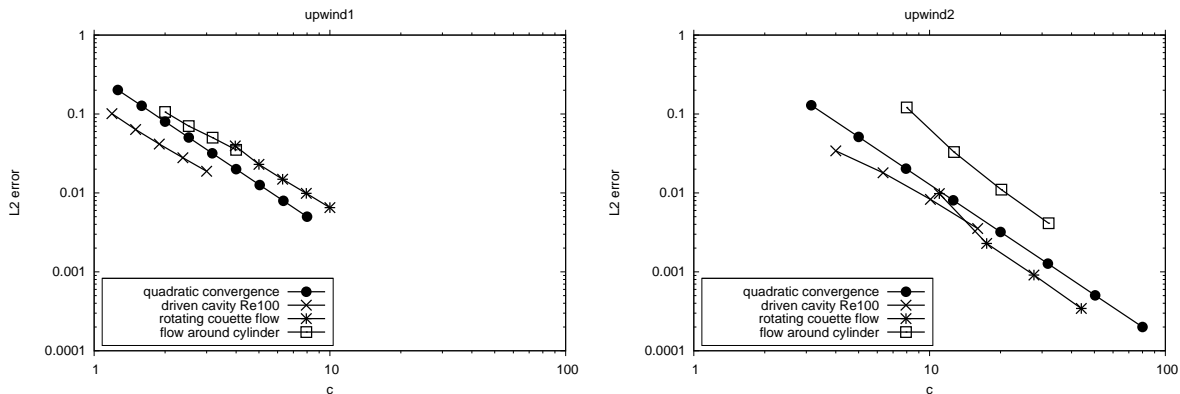


Figure 5: Comparison with theoretical (quadratic) convergence using fixed ratio $c \cdot h$

5.3 Convergence results for the nonlinear solver

In this section we present the results for treating the nonlinearity, hereby comparing the performance of the fixed point iteration with the Newton scheme. We provide the number of nonlinear iterations for *Driven Cavity* at $Re = 100$ and $Re = 600$ in Table 1, and in Table 2 results for the *Flow around Cylinder* benchmark at $Re = 2$ and $Re = 20$ which is more difficult due to the lower viscosity (see [24]). We find that the fixed point iteration is only able to cope with small nonlinearities without problems (the choice of bigger c also positively influences the solver). For higher values of Re , resp., lower viscosities some appropriate damping has to be applied to ensure convergence. Also, increasing the number of grid points N means more nonlinear iterations. In contrast, the Newton scheme performs well in all cases and needs only few iterations to reduce significantly the nonlinear defect (similar to [26]).

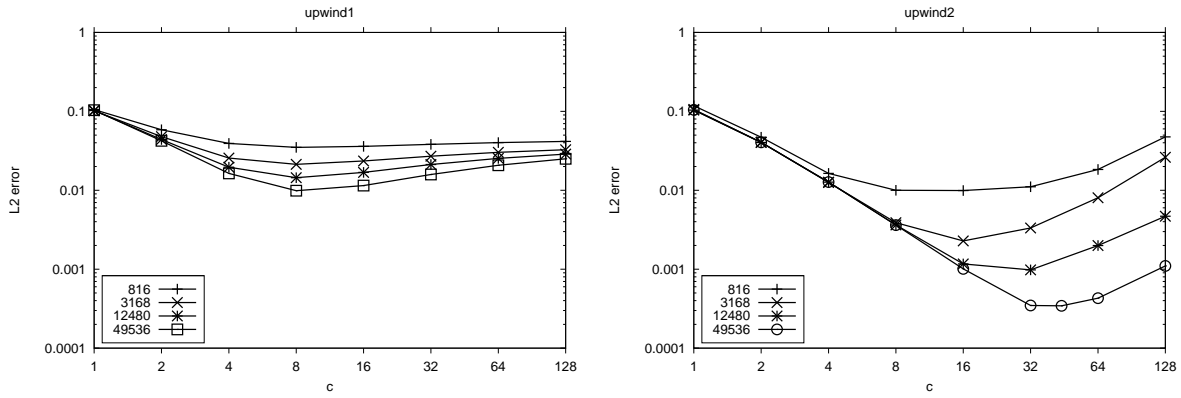


Figure 6: *Rotating Couette flow*, L_2 -error against c for upw1 (left), resp., upw2 (right)

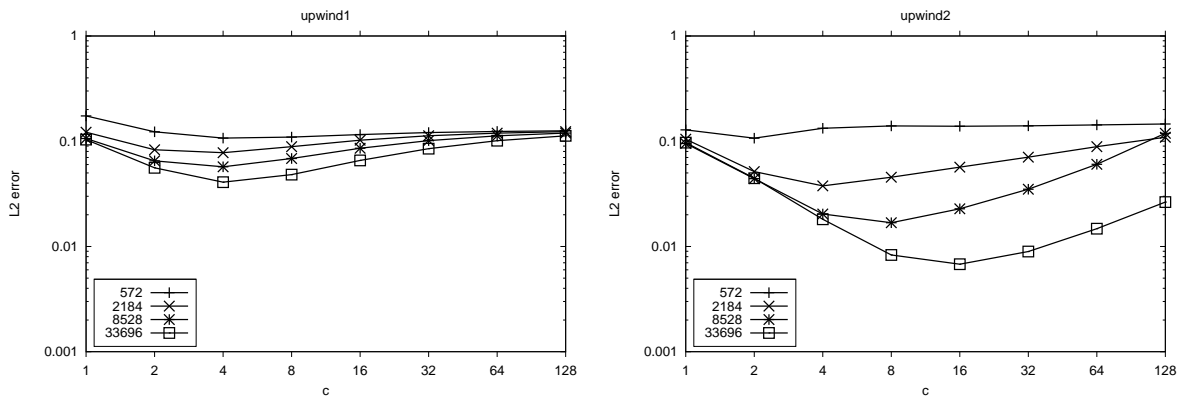


Figure 7: *Flow around Cylinder*, L_2 -error vs. c for upw1 (left), resp., upw2 (right)

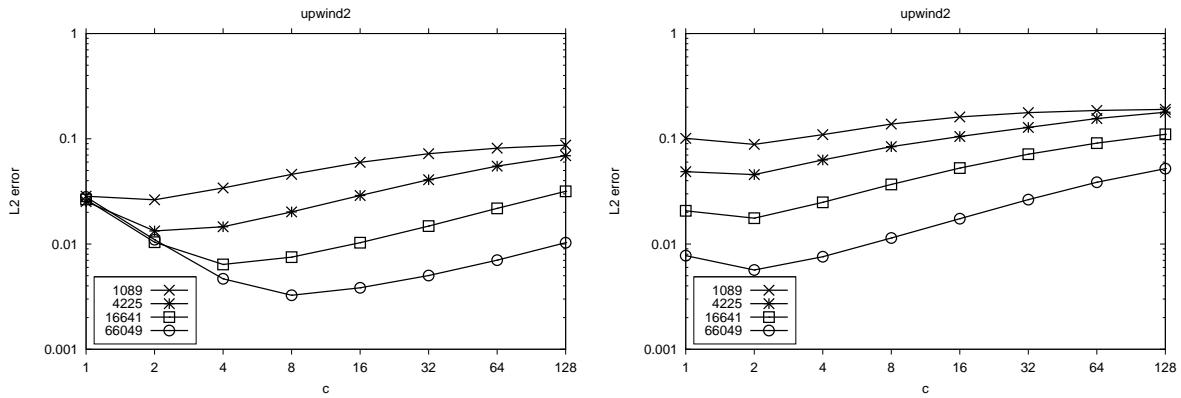


Figure 8: *Driven Cavity*, L_2 -error for upw2, $Re = 100$ (left), resp., $Re = 600$ (right)

grid points	Re 100		Re 600	
	fixed point	Newton	fixed point	Newton
		c=1		
289	15	4	27	6
1089	20	5	34	6
4225	24	5	59	8
		c=10		
289	8	4	21	4
1089	12	4	21	5
4225	17	4	36	6
		c=100		
289	5	3	22	3
1089	7	3	22	3
4225	10	4	22	4

Table 1: *Driven Cavity* with upw2: No. of iterations to reduce the nonlinear defect by 10^{-6}

grid points	Re 2		Re 20	
	fixed point	Newton	fixed point	Newton
		c=1		
572	13	4	22	5
2184	15	4	23	5
8528	16	4	23	5
		c=10		
572	11	4	20	4
2184	14	4	26	5
8528	16	4	46	5
		c=100		
572	6	3	20	3
2184	11	4	21	4
8528	15	4	118	6

Table 2: *Flow around cylinder* with upw2: No. of iterations to reduce the defect by 10^{-6}

5.4 Convergence results for the linear solver

We test the described linear solvers and preconditioners for all our configurations at various Reynolds numbers. The direct stationary solution of (3) proved to be too difficult to solve for the simple Richardson solver for interesting configurations. In this section we will focus therefore on presenting the results for the used Krylov-space methods, starting with the *Driven Cavity* configuration at $Re = 10$. More details can be found in [12].

First we provide the number of iterations to gain 6 digits for the linear defect using a preconditioned Bi-CGSTAB (see [30]) solver (see Table 3). It gives very good results in those fields our preconditioners were made for: First, at $c = 1$ with "tr-pre", the very mild dependence of h for transport dominated problems due to the almost exact preconditioner is shown on all levels. Second, for small numbers of grid points N with "bl-jac" preconditioning, this preconditioner is stable with increasing c , however, the results are highly level dependent. Unfortunately the convergence behaviour can fail unexpectedly, even for moderate configurations which is known for the Bi-CGSTAB scheme. Therefore, we also implemented a GMRES (see [23]) solver, taking advantage of its more monotone convergence behaviour. It still provides good results for transport dominated configurations, resp., for small N . Additionally, we get useful information about moderate and extreme configurations. While it is obvious that the separate use of "tr-pre" and "bl-jac" is not satisfactory, because it does not combine the advantages of both, the results (see Table 4) show the expected advantageous behaviour of the (preconditioned) GEF approach. For $c = 1$, the GEF without

preconditioning still yields very good convergence on all levels. With our diagonal preconditioning $\text{GEF}(\backslash)$ we gain additionally robustness w.r.t. high values of c . This means that our preconditioned GEF combines two positive effects, which is due to our special discretization technique. Nevertheless it remains the dependence on h , resp., N , for moderate and large values of c . That is why we use multigrid methods to overcome this last drawback.

grid points	upw1				upw2			
	tr-pre	bl-jac	GEF	$\text{GEF}(\backslash)$	tr-pre	bl-jac	GEF	$\text{GEF}(\backslash)$
c=1								
81	28	71	21	18	26	182	20	17
289	30	135	20	21	36	1000	23	20
1089	44	1000	24	30	43	1000	30	26
4225	52	1000	39	39	54	1000	41	36
c=10								
81	110	56	82	51	95	62	110	44
289	100	229	161	88	96	179	148	164
1089	111	716	153	133	141	1000	1000	1000
4225	137	1000	1000	351	143	1000	199	937
c=100								
81	1000	75	1000	60	1000	76	1000	92
289	1000	1000	1000	194	1000	349	1000	550
1089	1000	1000	1000	833	1000	1000	1000	1000
4225	1000	1000	1000	1000	1000	1000	1000	1000

Table 3: *Driven Cavity* at $Re = 10$, No. of iterations to gain 6 digits, using Bi-CGSTAB

grid points	upw1				upw2			
	tr-pre	bl-jac	GEF	$\text{GEF}(\backslash)$	tr-pre	bl-jac	GEF	$\text{GEF}(\backslash)$
c=1								
81	42	91	33	28	40	146	34	28
289	48	178	36	31	49	334	37	32
1089	56	336	40	45	63	716	44	42
4225	67	718	49	53	83	1000	56	53
c=10								
81	107	66	96	51	109	81	106	55
289	130	154	128	88	138	180	144	90
1089	158	351	158	129	164	423	182	147
4225	190	771	188	174	197	935	209	194
c=100								
81	235	67	211	67	227	78	246	77
289	360	170	407	155	409	203	502	182
1089	524	400	744	340	619	539	1000	461
4225	714	950	1000	689	838	1000	1000	1000

Table 4: *Driven Cavity* at $Re = 10$: No. of iterations to gain 6 digits, using GMRES

5.5 Monolithic multigrid solver for LBE

We have shown already that the GEF solver performs well for configurations with dominating transport operator, independent of the number of unknowns. Additionally, it becomes robust against large values of c with the block-diagonal preconditioning, but with level-dependent convergence rates for the relevant, intermediate range. That is why we implemented a prototypical version of the multigrid method. First,

we show the results for the *Driven Cavity* problem again at $Re = 10$ to compare with Table 4. For this purpose we had to implement grid-transfer operators: We apply linear and quadratic prolongation for the 1st and 2nd order discretization, respectively. However, so far we use only constant restriction, so that further optimization is feasible. Moreover, the use of multigrid as preconditioner in GMRES shows better convergence behaviour compared to the traditional multigrid algorithm. The rates for the latter variant are unsatisfactory, especially if only few smoothing steps are applied. This behaviour is comparable to [31] and shows the advantage of Krylov-space methods compared to "simple iterative schemes" even in a multigrid framework.

In Tables 5-6, we show the results for the GMRES with multigrid as preconditioner performing 1 cycle with 8 smoothing steps. We additionally compare the multigrid rates with the previous single grid results with *fixed* GMRES restart level of 16. We have to keep in mind that the GMRES algorithm stores all iteration vectors which needs a considerable amount of memory for large grids. The results show (almost) independence of the refinement level for the multigrid (even some improvement for finer levels), while the plain GEF results degrade for the case $c = 100$ as expected. The GEF(\backslash) with block-diagonal preconditioning yields significant improvement here, also for a higher Reynolds number of $Re = 100$ (see Table 6). If we concentrate on the moderate range of $c = 10$, we see that even without this stabilization the rates are quite good, and our tests indicate further improvement when more smoothing steps s are applied. However, increasing s does not improve the rates as much as expected, which means some slight aberration from common multigrid theory which might be due to the non-optimal restriction. The rates seem to improve like $O(\frac{1}{\sqrt{s}})$ rather than $O(\frac{1}{s})$ (see [12] for details). Moreover, we are not yet completely satisfied with the values for the 2nd order upwinding, but expect further improvement after implementing a quadratic restriction operator.

grid points	upw1				upw2				
	single grid		multigrid		single grid		multigrid		
	GEF	GEF(\backslash)	GEF	GEF(\backslash)	GEF	GEF(\backslash)	GEF	GEF(\backslash)	
	c=1								
289	0.06	0.05	0.05	0.04	0.05	0.08	0.09	0.07	
1089	0.12	0.08	0.07	0.06	0.05	0.07	0.14	0.12	
4225	0.13	0.14	0.13	0.16	0.13	0.16	0.14	0.17	
	c=10								
289	0.42	0.24	0.17	0.14	0.44	0.32	0.23	0.21	
1089	0.46	0.43	0.15	0.14	0.51	0.46	0.15	0.17	
4225	0.48	0.51	0.11	0.12	0.55	0.53	0.11	0.14	
	c=100								
289	0.73	0.43	0.37	0.19	0.80	0.51	0.61	0.33	
1089	0.83	0.64	0.28	0.19	0.89	0.75	0.51	0.39	
4225	0.89	0.79	0.22	0.18	0.94	0.88	0.38	0.35	

Table 5: *Driven Cavity* at $Re = 10$, convergence rates for multigrid as preconditioner in GMRES

5.6 Results for stationary flow around a 2D cylinder

Finally we use our "best" solver configuration within our FeatLBE code to solve the *Flow around Cylinder* benchmark at $Re = 20$ (see plots for the pressure and x -component of the velocity field in Fig. 9). By line integration of the stress acting on the cylinder one can obtain drag and lift values which are compared with the reference results from [24] (5.5795 and 0.0106, respectively, and $\Delta p = 0.1175$ for the pressure difference between front and back of the circle). In the LBM, the stress is usually obtained by evaluating the nonequilibrium distributions $f_i^{neq} = f_i - f_i^{eq}$ (see [19]). Results in Table 8 confirm that this method yields better results than the force evaluation based on point values of p and the gradient of \mathbf{u} , especially on high refinement levels.

We compare the performance of our new code with FeatFlow [27], using the same coarse mesh, but converting each quadrilateral into 4 triangular elements (see Fig. 4). Using again the monolithic, stationary approach

grid points	upw1				upw2				
	single grid		multigrid		single grid		multigrid		
	GEF	GEF(\)	GEF	GEF(\)	GEF	GEF(\)	GEF	GEF(\)	
	c=1								
289	0.41	0.29	0.20	0.16	0.48	0.33	0.27	0.23	
1089	0.51	0.44	0.17	0.15	0.55	0.46	0.18	0.19	
4225	0.54	0.46	0.08	0.08	0.59	0.54	0.14	0.15	
	c=10								
289	0.77	0.50	0.43	0.23	0.82	0.51	0.61	0.35	
1089	0.84	0.65	0.28	0.20	0.91	0.78	0.51	0.40	
4225	0.90	0.81	0.24	0.19	0.94	0.89	0.42	0.39	
	c=100								
289	0.89	0.64	0.73	0.24	0.91	0.71	0.87	0.46	
1089	0.94	0.75	0.64	0.24	0.97	0.87	0.85	0.52	
4225	0.97	0.86	0.53	0.22	0.99	0.94	0.80	0.55	

Table 6: *Driven Cavity at $Re = 100$* , convergence rates for multigrid as preconditioner in GMRES

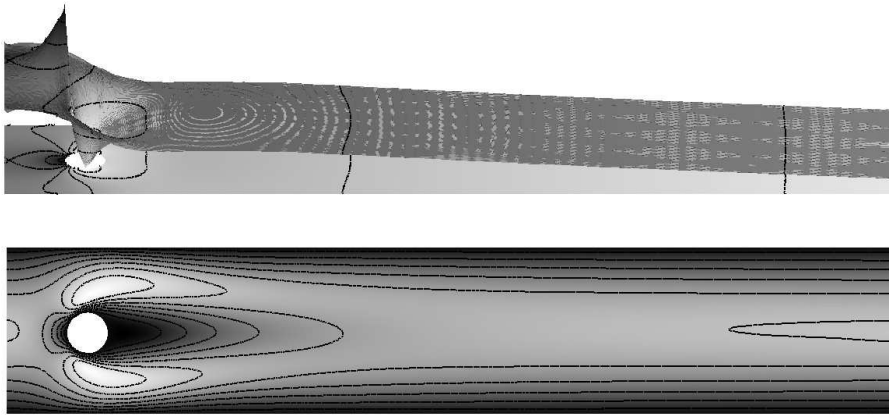


Figure 9: *Flow around Cylinder at $Re = 20$* , pressure and x-component of velocity

for this challenging problem, we start from the Level-1 solution (by interpolation) and use the same stopping criteria, i.e 10^{-8} for the nonlinear defect, resp., 10^{-4} linear gain each. The total number of unknowns is higher for FeatLBE because of $9 \times f_i$ per node in the D2Q9 model, nevertheless the CPU times (omitting the coarse grid solver, see Table 7) are already quite comparable to the highly optimized CFD code FeatFlow. However, the results strongly depend on the kind of discretization and stabilization of the convective terms (upwinding vs. edge-oriented EOFEM (see [29] for details) so that we expect a huge potential for the future by using improved numerical methods for LBE, too. At the moment, FeatLBE needs similarly few nonlinear steps and linear multigrid sweeps of the W-cycle with 16 smoothing steps. The rates for $c = 2$ are decreasing compared to $c = 1$, while the accuracy is increasing; however, findings from Section 5.2 indicate $c_{opt} < 5$ on the highest refinement level.

Finally, in order to improve the results, we construct a highly adapted grid which is easily accomplished using triangular elements (see Fig. 10). We obtain good results for drag and lift, and significant improvements for the results for Δp with less unknowns. With this methodology, mainly based on the described monolithic approach with the generalized equilibrium formulation, we also expect for the FeatLBE code even better linear rates for nonstationary flow at higher Re numbers, if we use high order time-stepping schemes in a fully implicit approach; the corresponding results will be given in a forthcoming paper.

FeatFlow		Upwind				EOFEM			
grid points	total unkn.	drag	lift	NL/AVMG	CPU	drag	lift	NL/AVMG	CPU
16848	42016	5.7460	0.0070	7/6	47	5.5803	0.0101	3/7	32
66976	167232	5.6196	0.0103	6/5	145	5.5789	0.0104	3/6	125
267072	667264	5.5882	0.0108	5/5	443	5.5793	0.0106	3/7	613
FeatLBE		c=1				c=2			
grid points	total unkn.	drag	lift	NL/AVMG	CPU	drag	lift	NL/AVMG	CPU
4264	38376	5.6676	0.0413	4/8	31	5.6490	0.0448	4/11	45
16848	151632	5.5287	0.0121	3/7	139	5.5403	0.0123	3/11	225
66976	602784	5.5398	0.0102	3/7	774	5.5863	0.0103	3/12	1308

Table 7: FeatFlow vs. FeatLBE results for *Flow around Cylinder*. Drag and lift coefficient, Nonlinear (NL)/Average multigrid sweeps (AVMG) and CPU time

grid points	p, \mathbf{u} force eval.			Δp	f^{neq} force eval.	
	drag	lift	drag		lift	
c=1						
2948	5.9034	0.0451	0.1240	5.6422	0.0466	
11592	5.9151	0.0209	0.1188	5.5191	0.0226	
45968	6.1069	0.0105	0.1177	5.5228	0.0128	
183072	6.3378	0.0075	0.1174	5.5284	0.0105	
c=2						
2948	6.0262	0.0937	0.1314	5.9606	0.0938	
11592	5.7718	0.0310	0.1209	5.6453	0.0315	
45968	5.7835	0.0134	0.1184	5.5801	0.0143	
183072	5.8641	0.0097	0.1178	5.5695	0.0109	
c=4						
2948	6.2414	0.1266	0.1398	6.2388	0.1267	
11592	5.7711	0.0404	0.1231	5.7388	0.0405	
45968	5.6735	0.0160	0.1188	5.6080	0.0163	
183072	5.6857	0.0110	0.1178	5.5812	0.0114	

Table 8: Drag and lift coefficient and pressure difference on highly adapted grid

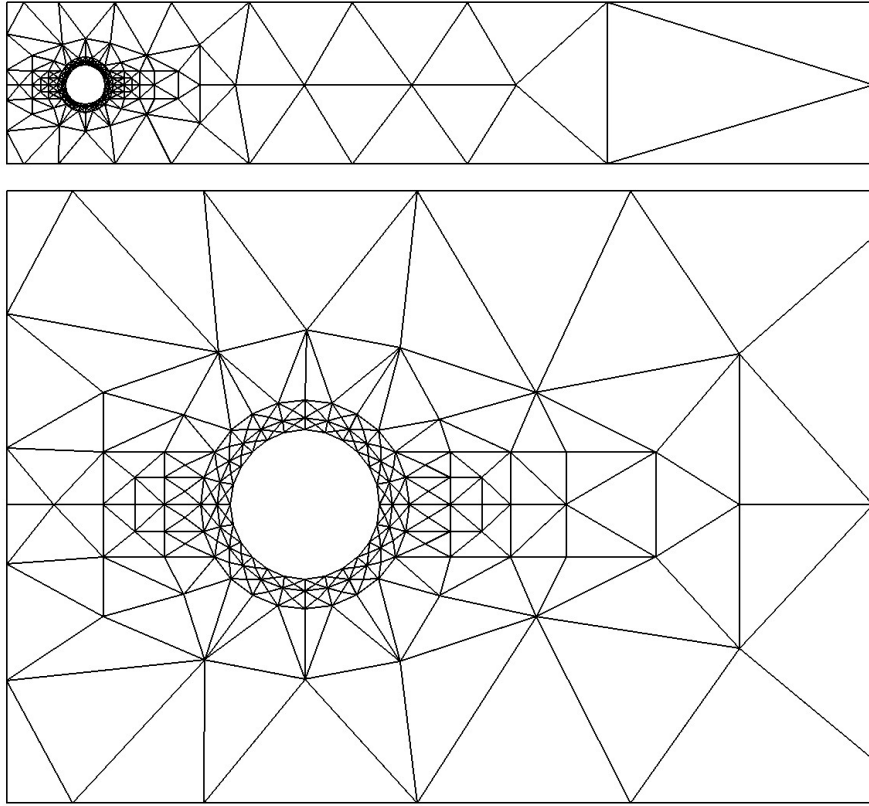


Figure 10: Highly adapted triangular grid for the *Flow around Cylinder* configuration

6 Conclusion and outlook

We have introduced a new approach for the Lattice Boltzmann equation (LBE) by applying highly sophisticated, modern techniques from Numerics for PDE: The implicit discretization of both, collision and advection, on general (unstructured) meshes is supposed to overcome the crucial CFL restriction of standard LB methods. We have shown that it is possible to treat the LBE in a more general way as a semi-discretized integro-differential equation, and thus to carry over solution techniques that have been successful in the field of radiative transfer problems. A new GEF reformulation of the LBE was derived which can combine the efficiency of direct transport solvers on the one hand with special preconditioning for collision dominated cases on the other hand leading to a very efficient and robust monolithic solver. The range of problem parameters was numerically analyzed to obtain a rule of thumb for an appropriate approximation to c_{opt} for various flow problems at different Reynolds numbers so that optimal approximation results can be obtained. Finally the performance was greatly improved using a multigrid algorithm and the achieved convergence rates are independent of the problem size. The ongoing research is concerned with an analysis of the MRT model and adjusting the solvers and preconditioners for the modified collision operator. We expect to get further understanding of the quality of our approximated solution, depending on the (variable) relaxation rates and choice of parameter c , and its interplay with corresponding preconditioning strategies which will be part of a forthcoming paper. Therein, we will also focus on unsteady flow problems, especially the *Flow around Cylinder* benchmark at $Re = 100$, and higher order time stepping schemes with large, resp., adaptive choice of Δt .

References

- [1] Bardow, A., Karlin, I. V., Gusev, A. A.: General characteristic-based algorithm for off-lattice Boltzmann simulations. *Europhys Letters* **75**, No. 3, 434–440 (2006)
- [2] Deo, N.: *Graph Theory with Applications to Engineering and Computer Science*. Prentice Hall International, 222–333 (1974)
- [3] d’Humières, D.: Generalized lattice-Boltzmann equations. In *Rarefied Gas Dynamics: Theory and Simulations* (ed. B.D. Shizgal and D.P. Weaver), *Prog Aeronaut Astronaut* **159**, 450–458 (1992)
- [4] d’Humières, D., Ginzburg, I., Krafczyk, M., Lallemand, P., Luo, L.-S.: Multiple-relaxation-time lattice Boltzmann models in three-dimensions. *Philos Trans R Soc Lond* **360**, 437–451 (2002)
- [5] d’Humières, D., Bouzidi, M., Lallemand, P.: Thirteen-velocity three-dimensional lattice Boltzmann model. *Phys Rev E* **63**, 066702 (2001)
- [6] Düster, A., Demkowicz, L., Rank, E.: High-order finite elements applied to the discrete Boltzmann equation. *Int J of Num Meth in Eng* **67**, 1094–1121 (2006)
- [7] Frisch U., d’Humières, D., Hasslacher, B., Lallemand, P.: Lattice Gas Hydrodynamics in Two and Three Dimensions. *Complex Systems* **1**, 75–136 (1987)
- [8] He, X., Luo, L. S.: Lattice Boltzmann model for the incompressible Navier-Stokes equation. *J of Stat Phys* **88**, 927–944 (1997)
- [9] Guo, Z., Zhao, T. S.: Explicit finite-difference lattice Boltzmann method for curvilinear coordinates. *Phys Rev E* **67**, 066709 (2003)
- [10] Hübner, T.: *Spezielle Diskretisierungs- und Lösungsmethoden für Integro-Differentialgleichungen am Beispiel der Strahlungstransportgleichung*. Diploma Thesis www.mathematik.uni-dortmund.de/lsvii/static/showpdf/Huebner2005.pdf, Dortmund (2005)
- [11] Hübner, T., Turek, S.: An efficient and accurate short-characteristics solver for radiative transfer problems. *Computing* **81**, 281–296 (2007)
- [12] Hübner, T.: *Monolithische Lösungsmethoden für die Lattice-Boltzmann Gleichung*. PhD Thesis, to appear (2009)
- [13] Hübner, T., Turek, S.: Efficient monolithic Lattice Boltzmann solver for nonstationary flow problems. to appear (2009)
- [14] Inamuro, T., Yoshino, M., Ogino, F.: Accuracy of the lattice Boltzmann method for small Knudsen number with finite Reynolds number. *Phys Fluids* **9**, 3535 (1997)
- [15] Junk, M.: LBM-discrete dynamics and finite difference method. *Proceedings of the GAMM Workshop "Discrete Modelling and Discrete Algorithms in Continuum Mechanics"*, Logos, Berlin (2001)
- [16] Ladd, A.: Numerical simulations of particulate suspensions via a discretized Boltzmann equation. *Journal of Fluid Mechanics* **271**, 285–309 (1994)
- [17] Lallemand, P., Luo, L. S.: Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys Rev E* **61**, No. 6, 6546–6562 (2000)
- [18] Mavriplis, D.J.: Multigrid solution of the steady-state Lattice Boltzmann equation. *Computer and Fluids* **35**, 573–591 (2006)
- [19] Mei, R., Yu, D., Shyy, W., Luo, L.-S.: Force evaluation in the Lattice Boltzmann method involving curved geometry. *Phys Rev E*, **65**, 041203 (2002)
- [20] Noble, D., Holdych, D.: Full Newton Lattice Boltzmann Method for Time-Steady Flows using a Direct Linear Solver. *Int J of Mod Phys C* **18**, No. 4, 652–660 (2007)

- [21] Qian, Y.H., d’Humières, D., Lallemand, P.: Lattice BGK models for Navier-Stokes equation. *Europhys Letters* **17**, 479–484 (1992)
- [22] Reider, M., Sterling, J.: Accuracy of discrete velocity BGK models for the simulation of the incompressible Navier Stokes equations. *Computers and Fluids* **24**, 459–467 (1995)
- [23] Saad, Y., Schultz, M.H.: GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J Sci and Stat Comput* **7**, No. 3, 856–869 (1986)
- [24] Schäfer, M., Turek, S.: Benchmark computations of laminar flow around cylinder. *Notes on Numerical Fluid Mechanics* **52**, 547–566 (1996)
- [25] Tölke, J., Krafczyk, M., Rank, E.: A Multigrid-Solver for the Discrete Boltzmann-Equation. *J Stat Phys* **107**, No. 1/2, 573–591 (2002)
- [26] Tölke, J., Krafczyk, M., Schulz, M., Rank, E., Berrios, R.: Implicit discretization and non-uniform mesh refinement approaches for FD discretizations of LBGK Models. *International Journal of Modern Physics C* **9**, No. 8, 1143–1157 (1998)
- [27] Turek, S.: FEATFLOW Finite Element software for the incompressible Navier-Stokes equations: User Manual (www.featflow.de). Release 1.2, University of Dortmund, (2000)
- [28] Turek, S., A Generalized Mean Intensity Approach for the Numerical Solution of the Radiative Transfer Equation. *Computing* **54**, 27–38 (1995)
- [29] Turek, S., Ouazzi, A.: Unified edge-oriented stabilization of nonconforming FEM for incompressible flow problems: Numerical investigations. *J Num Math* **14**, No. 4, 299–322 (2007)
- [30] Van der Vorst, H.: BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J Sci Stat Comput* **13**, 631–644 (1992)
- [31] Wobker, H.; Turek, S.: Numerical Studies of Vanka-type Smoothers in Computational Solid Mechanics. *Adv Appl Math Mech* **1**, 29–55 (2009)
- [32] Zou, Q.; He, X.: On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids* **9**, No. 6, 1591–1598 (1997)