High-order finite elements applied to the discrete Boltzmann equation

Alexander Düster\textsuperscript{1,*,†}, Leszek Demkowicz\textsuperscript{2} and Ernst Rank\textsuperscript{1}

\textsuperscript{1}Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, TU München, Arcisstr. 21, 80290 München, Germany
\textsuperscript{2}The Institute for Computational Engineering and Sciences, The University of Texas at Austin, 105 Austin, TX 78712, U.S.A.

SUMMARY

A discontinuous Galerkin approach for solving the discrete Boltzmann equation is presented, allowing to compute approximate solutions for fluid flow problems. Based on a two-dimensional high-order finite element and an explicit Euler time stepping scheme, the D2Q9 model is discretized and the results are compared to the exact solution of the Navier–Stokes equation. Four numerical examples are considered, including stationary and instationary problems with curved boundaries. It is demonstrated that the proposed method allows to obtain the desired, highly efficient exponential convergence. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: discrete Boltzmann equation; fluid dynamics; $p$-FEM

1. INTRODUCTION

During the last two decades the lattice Boltzmann method (LBM) [1–4] has emerged as an attractive alternative to computational fluid dynamics (CFD) based on the Navier–Stokes (NS) equations. Recent benchmark computations for laminar flows [5] carrying out a detailed comparison of the LBM with academic and commercial state-of-the-art CFD solvers have demonstrated that a LB prototype is competitive even for incompressible transient laminar problems. For the weakly compressible case, the LBM was shown to have significant wall clock time advantage as compared to a commercial CFD code.

The LBM [2–4,6–8] can be interpreted as a first-order explicit upwind finite difference (FD) discretization of the discrete Boltzmann equation. In recent years, increasing effort has
been made to perform LB simulations on nonuniform grids [9–13] in order to control the discretization error more efficiently. As an alternative approach to the FD discretization of the discrete Boltzmann equation, finite volume (FV) schemes [14–19] have been developed, allowing to perform simulations on unstructured grids.

Concerning the finite element method (FEM), to the best knowledge of the authors, there have been only two publications addressing the discretization of the discrete Boltzmann equation. Lee and Lin [20] present a characteristic Galerkin FEM for the discrete Boltzmann equation to simulate fluid flows in complex geometries. Linear triangular and bilinear quadrilaterals are applied to discretize the D2Q9 model. The temporal discretization is based on a second-order FD-predictor-corrector method. Numerical examples demonstrate that the expected convergence rate can be observed. Due to the use of unstructured meshes, curved boundaries and high gradients in the exact solution can be efficiently resolved. Shi et al. [21] present a discontinuous Galerkin spectral element method for the D2Q9 model. Their spatial discretization applies Dubiner’s basis with polynomials of order up to $p = 8$. An explicit Adams–Bashforth scheme is chosen to perform the temporal discretization. A numerical example dealing with the flow past a circular cylinder demonstrates that the method allows to perform accurate computations on unstructured triangular meshes.

The aim of this paper is to contribute to the development of high-order finite element methods for the discrete Boltzmann equation. In contrast to the work of Shi et al. [21] we focus on the following topics. Two different methods for defining the velocity of the fluid at a wall are investigated with respect to their influence on the accuracy of high-order finite elements. Ladd’s [22, 23] and Inamuro et al.’s [24, 25] method for describing boundary conditions are compared by computing the shear flow between two parallel plates (Couette flow) as well as simulating a forced channel flow (plane Poiseuille flow). Considering the shear flow between two coaxial cylinders (rotating Couette flow), it is demonstrated that half of the computational domain can be accurately discretized with only one element when combining the blending function method with an increase in polynomial degree of the trial and test functions. Special emphasis is put on the accuracy and convergence rate of the method when performing a $p$-extension. The discretization error as well as the model error, i.e. the deviation from the exact NS solution are investigated.

The outline of the paper is as follows: In Section 2, the discrete Boltzmann equation based on the D2Q9 model is briefly summarized. The discretization based on a discontinuous Galerkin approach is presented in Section 3. Numerical examples presented in Section 4 investigate the accuracy and efficiency of the proposed discretization scheme for selected laminar flow problems for which an exact NS solution is available.

## 2. DISCRETE BOLTZMANN EQUATION

Our point of departure for fluid flow simulations is the discrete Boltzmann equation. We will consider the so-called D2Q9 model which will be briefly summarized in the following section.

### 2.1. The D2Q9 model

#### 2.1.1. Differential equation
The two-dimensional D2Q9 model stems from the Boltzmann equation where the complicated collision operator is approximated by the single-time relaxation
collision model proposed by Bhatnagar–Gross–Krook (BGK) [26]. The microscopic velocity space has been discretized, resulting in nine discrete velocities $e_i$, see Figure 1.

The underlying mathematical model is therefore described by a set of nine coupled partial differential equations

$$\frac{\partial f_i}{\partial t} + e_i \cdot \nabla f_i = -\frac{1}{\tau} (f_i - f_i^{(eq)}) + F_i \quad \text{in} \quad \Omega \subset \mathbb{R}^2, \quad i = 0, 1, 2, \ldots, 8$$  \hspace{1cm} (1)

where $f_i = f(x, e_i, t) \, (\text{kg/m}^3)$ is the particle distribution function associated with direction $i$ in the discrete velocity space. The right-hand side of Equation (1)

$$C_i = -\frac{1}{\tau} (f_i - f_i^{(eq)}) + F_i$$  \hspace{1cm} (2)

consists of the discrete counterpart of the BGK model, where $\tau(s)$ denotes the relaxation time and

$$F_i = \alpha F \cdot e_i$$  \hspace{1cm} (3)

$$\alpha = 1 \left/ \sum_{i=0}^{8} e_{ix}^2 \right. = 1 \left/ \sum_{i=0}^{8} e_{iy}^2 \right. = 1/6$$  \hspace{1cm} (4)

which corresponds to the contribution of a volume force $F (\text{kg/s}^2/\text{m}^2)$ in the $i$th direction. The discrete velocities $e_i \, (\text{m/s})$ are defined by

$$e_0 = 0$$  \hspace{1cm} (5)

$$e_i = c \cos \left( \frac{i - 1}{2} \pi \right) e_x + c \sin \left( \frac{i - 1}{2} \pi \right) e_y, \quad i = 1, 2, 3, 4$$  \hspace{1cm} (6)

$$e_i = \sqrt{2}c \cos \left( \frac{i - 4.5}{2} \pi \right) e_x + \sqrt{2}c \sin \left( \frac{i - 4.5}{2} \pi \right) e_y, \quad i = 5, 6, 7, 8$$  \hspace{1cm} (7)
where \( c \) (m/s) is a constant which has been set to unity and \( \mathbf{e}_x, \mathbf{e}_y \) denote the Cartesian basis vectors, see Reference [27]. The equilibrium distribution function

\[
f_i^{(eq)} = \rho w_i \left( 1 + \frac{3}{c^2} \mathbf{e}_i \cdot \mathbf{u} + \frac{9}{2c^2} (\mathbf{e}_i \cdot \mathbf{u})^2 - \frac{3}{2c^2} \mathbf{u} \cdot \mathbf{u} \right), \quad i = 0, 1, 2, \ldots, 8 \quad (8)
\]

where \( w_0 = 4/9, w_1 = w_2 = w_3 = w_4 = 1/9, w_5 = w_6 = w_7 = w_8 = 1/36 \), has been derived by using the low-Mach number approximation of the Maxwell–Boltzmann distribution. The macroscopic quantities, i.e., the density \( \rho \) (kg/m\(^3\)), velocity \( \mathbf{u} \) (m/s) and pressure \( p \) (kg/s\(^2\)/m) of the fluid are computed as

\[
\rho = \sum_{i=0}^{8} f_i \quad (9)
\]

\[
\mathbf{u} = \frac{1}{\rho} \sum_{i=0}^{8} f_i \mathbf{e}_i = \frac{1}{\rho} \sum_{i=1}^{8} f_i \mathbf{e}_i \quad (10)
\]

\[
p = c_s^2 \rho \quad (11)
\]

where \( c_s = c/\sqrt{3} \) (m/s) is the isothermal speed of sound. The kinematic viscosity \( \nu \) (m\(^2\)/s) of the fluid can be computed from the relaxation time

\[
\nu = c_s^2 \tau \quad (12)
\]

### 2.1.2. Boundary conditions.
To uniquely determine the solution of Equation (1), boundary conditions need to be defined. Considering the lattice Boltzmann method, there exists a wealth of publications dealing with different kinds of boundary conditions. There are two major difficulties:

The first problem arises due to the fact that one typically wants to define boundary conditions in terms of macroscopic quantities like velocity or pressure. However, these quantities do not enter the discrete Boltzmann equation directly, but they are linked to the \( f_i \)'s of Equation (1) by means of Equations (9)–(11). Therefore it is not straightforward to derive unique boundary conditions for the particle distribution functions \( f_i \), starting with those expressed in terms of the macroscopic quantities.

The second problem is related to the finite difference scheme applied in the lattice Boltzmann method. The finite difference grid nodes actually never coincide with the exact location of the no-slip condition which can be shown to be depending on the value of \( \tau \) in the case of single relaxation time models, see e.g. Reference [28]. While the second problem has been solved already in References [11, 13, 29] and is also not an issue with the present approach, we have no general solution for the first problem.

Two different kinds of boundary conditions are applied in the computation of the numerical examples presented in Section 4. Periodic boundary conditions are applied in order to reduce the size of the computational domain. Considering, for example, the task of prescribing periodic boundary conditions for an infinitely long channel, we require that the velocity on the left-hand side (West) is equal to the velocity on the right-hand side (East) \( \mathbf{u}^{W} = \mathbf{u}^{E} \), see Figure 2.
We enforce it by setting

$$f_i^W = f_i^E, \quad i = 0, 1, 2, \ldots, 8$$

(13)

As a further type of boundary conditions we consider moving walls, i.e. solid boundaries which move with a prescribed velocity. In Figure 3, two different situations are depicted. On the left-hand side a wall is located in the north, whereas on the right-hand side a wall is located in the south. In order to describe the boundary condition at a wall, two different situations are
distinguished, the inflow and the outflow, i.e.
\begin{align}
\mathbf{e}_i \cdot \mathbf{n} < 0 & \quad \text{on } \Gamma_{\text{in}}(\Omega) \quad (14) \\
\mathbf{e}_i \cdot \mathbf{n} \geq 0 & \quad \text{on } \Gamma_{\text{out}}(\Omega) \quad (15)
\end{align}

Since Equation (1) form a system of first-order hyperbolic equations, we prescribe only inflow boundary conditions. The basic idea is to define the inflow conditions for $f_i$'s in such a way that the resulting velocity of the fluid equals the velocity of the wall. In the lattice Boltzmann context, there are different approaches which have been proposed in the literature. Considering a north wall with a prescribed velocity $\mathbf{u}_w$, Ladd [22, 23] suggests to set up the inflow boundary conditions in the form
\begin{align}
f_3^* &= f_2 - 2 \rho_0 \frac{u_w}{c_s^2} \mathbf{e}_2 \cdot \mathbf{u}_w \\
f_7^* &= f_3 - 2 \rho_0 \frac{u_w}{c_s^2} \mathbf{e}_5 \cdot \mathbf{u}_w \\
f_8^* &= f_6 - 2 \rho_0 \frac{u_w}{c_s^2} \mathbf{e}_6 \cdot \mathbf{u}_w \quad (16)
\end{align}

based on the required change of momentum. If the velocity of the wall is assumed to be zero, i.e. $\mathbf{u}_w = 0$, Equation (16) reduce to the well-known bounce-back scheme. However, the drawback of the bounce-back scheme is that it introduces a slip velocity which is in conflict with the prescribed velocity $\mathbf{u}_w = 0$. In order to overcome this problem, Inamuro et al. [24, 25] proposed a non-slip boundary condition for lattice Boltzmann simulations. The idea is to assume the inflow to be an equilibrium distribution function with a counter slip velocity which is determined in such a way that the fluid velocity at the wall is equal to the wall velocity.

Considering again a wall located in the north, the inflow conditions read as follows:
\begin{align}
f_3^* &= \frac{1}{2} \rho'(1 - 3v_w + \frac{9}{2} v_w^2 - \frac{9}{2} ((u_w + u')^2 + v_w^2)) \\
f_7^* &= \frac{1}{36} \rho'(1 + 3(-u_w - u' - v_w) + \frac{9}{2} (-u_w - u' - v_w)^2 - \frac{9}{2} ((u_w + u')^2 + v_w^2)) \\
f_8^* &= \frac{1}{36} \rho'(1 + 3(u_w + u' - v_w) + \frac{9}{2} (u_w + u' - v_w)^2 - \frac{9}{2} ((u_w + u')^2 + v_w^2)) \\
\rho_w &= \frac{1}{1 - v_w} (f_0 + f_1 + f_3 + 2(f_6 + f_2 + f_5)) \\
\rho' &= 6 \rho_w u_w + f_2 + f_6 + f_5 \\
u' &= \frac{1}{1 + 3v_w} \left(6 \rho_w u_w - \frac{f_1 - f_3 - f_5 - f_6}{\rho'} - u_w - 3u_w v_w \right) \quad (17)
\end{align}

2.1.3. Initial conditions. In addition to setting up the boundary conditions, we have to define initial conditions for the system of Equation (1). Assuming that the mean density $\bar{\rho}$ and velocity $\bar{u}$ of the fluid at time $t = 0$ may approximately be computed, the nine initial distribution
functions $\tilde{f}_i$ are determined by means of the equilibrium function $f^{(eq)}_i(\tilde{\rho}, \tilde{u})$.

\begin{align}
\tilde{\rho} &= \rho(x, y, t = 0) \\
\tilde{u} &= u(x, y, t = 0) \\
\tilde{f}_i &= f^{(eq)}_i(\tilde{\rho}, \tilde{u}), \quad i = 0, 1, 2, \ldots, 8
\end{align}

Initial gradients can also be considered by computing the corresponding first-order non-equilibrium contributions as described in Reference [30], see also Section 4.4.

3. DISCRETIZATION OF THE D2Q9 MODEL

3.1. Discontinuous Galerkin method

The starting point of the finite element formulation is a discontinuous Galerkin approximation of Equation (1) which reads

\begin{equation}
\int_{\Omega_e} \phi \frac{\partial f_i}{\partial t} \, d\Omega = -\int_{\Omega_e} \phi \mathbf{e}_i \cdot \nabla f_i \, d\Omega + \int_{\Gamma_{in}(\Omega_e)} \phi [f_i] \mathbf{e}_i \cdot \mathbf{n} \, d\Gamma + \int_{\Omega_e} \phi C_i \, d\Omega \quad i = 0, 1, 2, \ldots, 8 \quad \forall \phi
\end{equation}

where $\phi$ is a test function and

\begin{align}
[f_i] &= f_i^+ - f_i^- \\
f_i^\pm &= \lim_{\varepsilon \to 0} f_i(x \pm \varepsilon \mathbf{e}_i) \\
f_i^-(x) &= f_i^+(x) \quad \text{if} \ x \in \Gamma_{in}(\Omega) \\
\mathbf{e}_i \cdot \mathbf{n} &\geq 0 \quad \text{on} \ \Gamma_{out}(\Omega_e) \\
\mathbf{e}_i \cdot \mathbf{n} &< 0 \quad \text{on} \ \Gamma_{in}(\Omega_e)
\end{align}

Equation (21) is formulated for each of the $e = 1, 2, 3, \ldots, n_{el}$ non-overlapping elemental domains $\Omega_e$ of the whole computational domain $\Omega = \bigcup_{e=1}^{n_{el}} \Omega_e$ separately, since the approximation may be discontinuous across the elemental boundaries. Therefore, we have two different values of $f_i$ at each point of an elemental boundary, where $f_i^+ = \lim_{\varepsilon \to 0} f_i(x + \varepsilon \mathbf{e}_i)$ is the limit 'from above' and $f_i^- = \lim_{\varepsilon \to 0} f_i(x - \varepsilon \mathbf{e}_i)$ is the limit 'from below' which can be interpreted as the value of $f_i$ upstream the characteristic direction $\mathbf{e}_i$. The jump $f_i^+ - f_i^-$ is denoted by $[f_i]$. The elemental solutions are coupled via the boundary term

\begin{equation}
\int_{\Gamma_{in}(\Omega_e)} \phi [f_i] \mathbf{e}_i \cdot \mathbf{n} \, d\Gamma
\end{equation}
HIGH-ORDER FINITE ELEMENTS APPLIED TO THE DISCRETE BOLTZMANN EQUATION

If the elemental boundary happens to be a part of the inflow boundary \( \Gamma_{in}(\Omega) \), \( f_i^- (\mathbf{x}) \) is replaced by \( f_i^+ (\mathbf{x}) \) which is either determined by Ladd's or Inamuro's boundary condition, see Section 2.1.2. If we consider, for example, a finite element with a wall located in the north as depicted on the left-hand side of Figure 2, we have

\[
[f_i] = \begin{cases} 
  f_4 - f_4^+ & \text{if } i = 4 \\
  f_7 - f_7^+ & \text{if } i = 7 \\
  f_8 - f_8^+ & \text{if } i = 8 \\
  0 & \text{else}
\end{cases} 
\]  
(28)

with \( f_4 = f_4^+ \), \( f_7 = f_7^+ \), \( f_8 = f_8^+ \) whereas \( f_4^*, f_7^*, f_8^* \) are determined either from Equation (16) or (17).

Periodic boundary conditions are applied in the strong sense, see Equation (13).

Initial conditions

\[
\bar{f}_i = f_i^{\text{eq}} (\bar{\rho}, \bar{\mathbf{u}}), \quad i = 0, 1, 2, \ldots, 8
\]  
(29)

are computed element-wise by performing the \( L_2 \)-projection of \( \bar{f}_i \) onto the finite element space, i.e. by minimizing

\[
F(\bar{f}_i) = \int_{\Omega_e} (f_i - \bar{f}_i)^2 \, d\Omega, \quad i = 0, 1, 2, \ldots, 8
\]  
(30)

where \( \bar{f}_i \) denotes the finite element approximation (32) of the particle distribution function associated with direction \( i \).

For a detailed discussion of the discontinuous Galerkin method for advection-dominated problems the reader is referred to References [31–33] and the literature listed therein.

3.2. Spatial discretization

In the following, a spatial discretization of Equation (21) is presented. Since the discontinuous Galerkin approach is applied, no attention to interelement continuity has to be paid. Each element may have a different polynomial degree with shape functions being discontinuous across the element interfaces. Therefore, it is sufficient to consider one elemental domain \( \Omega_e \) separately. The coupling to adjacent elements is realized through the elemental inflow boundary term (27), as discussed in the previous section.

The test and trial functions

\[
\phi \approx \hat{\phi} = k_{\text{nodes}} \sum_{k=1}^{k_{\text{nodes}}} N_k \hat{\phi}_k = \mathbf{N} \hat{\phi}
\]  
(31)

\[
f_i \approx \hat{f}_i = k_{\text{nodes}} \sum_{k=1}^{k_{\text{nodes}}} N_k \hat{f}_{i,k} = \mathbf{N} \hat{f}_i
\]  
(32)
are discretized by choosing the same set of shape functions $N_k$ which are conveniently stored in a matrix

$$\mathbf{N} = [N_1 \ N_2 \ N_3 \ \cdots \ N_{k_{\text{nodes}}}]$$

(33)

The gradient of the trial function is determined by computing the derivatives of the shape functions

$$\nabla \hat{f}_i = \sum_{k=1}^{k_{\text{nodes}}} \nabla N_k \hat{f}_{i,k} = \mathbf{B} \hat{f}_i$$

(34)

where

$$\mathbf{B} = \begin{bmatrix}
\frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \cdots & \frac{\partial N_{k_{\text{nodes}}}}{\partial x} \\
\frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \cdots & \frac{\partial N_{k_{\text{nodes}}}}{\partial y}
\end{bmatrix}$$

(35)

denotes the so-called $\mathbf{B}$-matrix, collecting the derivates of the shape functions with respect to the global co-ordinates $x$, $y$. Inserting discretization (31)–(35) into Equation (21) we obtain

$$\int_{\Omega_e} \mathbf{N}^T \mathbf{N} \, d\Omega \, \frac{d\hat{\mathbf{f}}_i}{dt} = -\int_{\Omega_e} \mathbf{N}^T \mathbf{e}_i \cdot \mathbf{B} \hat{f}_i \, d\Omega + \int_{\Gamma_{\text{int}}(\Omega_e)} \mathbf{N}^T \mathbf{n} \cdot \mathbf{e}_i [\hat{\mathbf{f}}_i] \, d\Gamma + \int_{\Omega_e} \mathbf{N}^T \mathbf{C}_i \, d\Omega$$

(36)

which can be alternatively written as

$$\mathbf{M} \frac{d\hat{\mathbf{f}}_i}{dt} = \mathbf{L}(\hat{\mathbf{f}}_i)$$

(37)

where $\mathbf{M}$ denotes the so-called mass matrix and $\mathbf{L}(\hat{\mathbf{f}}_i)$ is the right-hand side including non-linearities which are introduced by the equilibrium distribution function, see Equations (2) and (8). Having performed the spatial discretization, a set of ordinary differential equations remains to be solved. Before we address the temporal discretization, a few words about the shape functions and mapping used in this approach are in order.

The element formulation is based on a quadrilateral, see Figure 4, and applies the shape functions proposed by Szabó and Babuška [34]. The shape functions can be classified into three groups.

1. Nodal or vertex modes: The nodal modes

$$N_{1,i}^{N_i}(\xi, \eta) = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta), \quad i = 1, \ldots, 4$$

(38)

are the standard bilinear shape functions, well known from the isoparametric four-noded quadrilateral element. $(\xi_i, \eta_i)$ denote the local coordinates of node $i$.

2. Edge or side modes: These modes are defined separately for each individual edge, they vanish at all other edges. The corresponding modes for edge $E_1$ read:

$$N_{1,i}^{E_1}(\xi, \eta) = \frac{1}{2}(1 - \eta)\phi_1(\xi), \quad i \geq 2$$

(39)
3. Internal modes: The internal modes

\[ N_{i,j}^{\text{int}}(\xi, \eta) = \phi_i(\xi)\phi_j(\eta), \quad i, j \geq 2 \]

are purely local and vanish along all edges of the quadrilateral element.

The indices \( i, j \) of the shape functions denote the corresponding polynomial degrees in the local directions \( \xi, \eta \). The one-dimensional functions

\[
\phi_j(\xi) = \frac{2j-1}{2} \int_{-1}^{1} L_{j-1}(x) \, dx
\]

\[
= \frac{1}{\sqrt{4j-2}} (L_j(\xi) - L_{j-2}(\xi)), \quad j = 2, \ldots
\]

are based on (integrated) Legendre polynomials

\[
L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad x \in (-1, 1), \quad n = 0, 1, 2, \ldots
\]

The tensor product space has been applied in this paper, resulting in \((p + 1)^2\) shape functions for a polynomial degree \( p \).

In low-order finite element analysis, the most frequently used mapping concept for the geometric description of the domain of computation is the application of isoparametric elements. The same shape functions are used for the approximation of the unknown solution and for the shape of the elements. Using elements of order \( p = 1 \) or 2 the boundary of the domain is therefore approximated by a polygonal or by a piecewise parabolic curve, respectively. As the mesh is refined, the boundary of the domain is approximated more and more accurately. When using the \( p \)-version, on the other hand, the mesh remains fixed. It is therefore important to model the geometry of the structure accurately with the fixed number of elements in use. This calls for a method which is able to describe complex geometries using only a few elements. Gordon and Hall [35, 36] proposed the so-called blending function method, which is usually applied when describing curved boundaries of \( p \)-version finite elements. Consider a quadrilateral element as shown in Figure 5 where edge \( E_2 \) is assumed to be part of a curved boundary. The shape of
edge $E_2$ is assumed to be defined by a parametric function $\mathbf{E}_2 = [E_{2x}(\eta), E_{2y}(\eta)]^T$, where $\eta$ is the local coordinate of the element. The transformation of the local coordinates $\xi = [\xi, \eta]^T$ into the global coordinates $\mathbf{x} = [x, y]^T = \mathbf{Q}^e = [Q_x^e(\xi, \eta), Q_y^e(\xi, \eta)]^T$ can be formulated by the two functions

\[
\begin{align*}
x &= Q_x^e(\xi, \eta) = \sum_{i=1}^{4} N_{i,1}(\xi, \eta) X_i + \left(E_{2x}(\eta) - \left(\frac{1-\eta}{2}X_2 + \frac{1+\eta}{2}X_3\right)\right) \frac{1+\xi}{2} \\
y &= Q_y^e(\xi, \eta) = \sum_{i=1}^{4} N_{i,1}(\xi, \eta) Y_i + \left(E_{2y}(\eta) - \left(\frac{1-\eta}{2}Y_2 + \frac{1+\eta}{2}Y_3\right)\right) \frac{1+\xi}{2}
\end{align*}
\]

where the first term corresponds to the standard bilinear mapping, which is familiar from the isoparametric concept for quadrilateral elements with $p = 1$. The second term takes the curved edge $E_2$ into account. Therefore, the bilinear mapping is augmented by the blended difference between the curve $\mathbf{E}_2 = [E_{2x}(\eta), E_{2y}(\eta)]^T$ and the straight line connecting the nodes $N_2$ and $N_3$. The blending term $(1+\xi)/2$ ensures that the opposite edge $E_4$—where $(1+\xi)/2 = 0$—is not affected by the curvilinear description of edge $E_2$. This concept can be readily extended to quadrilaterals where all four edges are curved. A detailed discussion of shape functions, Ansatz spaces and mapping techniques for elements of high order can be found in References [34, 37–39].

Remarks

(i) In this paper, a hierarchical basis for quadrilaterals proposed by Szabó and Babuška [34] has been applied. This hierarchical basis groups the shape functions into interior and exterior modes which permits efficient enforcement of continuity. However, when using a discontinuous Galerkin approach continuity at element interfaces is not an issue. Therefore, the split into nodal, edge and internal modes is not necessary. One could also simply use tensor products of one-dimensional (Legendre) polynomials, since the shape functions may be discontinuous across the element interfaces.
(ii) The shape functions (38)–(40) make use of integrated Legendre polynomials (41). This is to reduce the condition number of stiffness matrices, where the entries are integrals of products of derivatives of shape functions. If, however, an explicit time stepping scheme is applied to solve Equation (37), then the condition of the mass matrix is of interest. Therefore it might be more efficient to use a different set of shape functions like products of Legendre polynomials or Dubiner’s basis [21, 40–42]. Ideally, the mass matrix reduces to the identity matrix and no equation system has to be solved.

(iii) The blending function method allows to completely decouple the shape functions being used for the trial and test functions, from those used for the mapping. Therefore, the mapping concept may be either subparametric, isoparametric or superparametric, see e.g. Reference [43]. When using a superparametric concept, it is known from finite element applications for solid mechanics problems that rigid body modes or equivalently zero eigenvalues may not be represented correctly. In Reference [39, 44] this drawback was investigated in detail. Summarizing these investigations, one can conclude that if the geometry is ‘smooth’ and can be represented satisfactorily by means of polynomials, the error due to an inaccurate representation of rigid body modes (or zero eigenvalues) decreases very rapidly for increasing orders of the Ansatz space. Note that the exact representation of rigid body modes corresponds to the question whether polynomials of degree $p = 1$ can be represented in the global coordinate system by the element in use.

3.3. Temporal discretization

In this paper, a first-order forward Euler time discretization

$$\mathbf{M} \frac{\mathbf{\Delta} \mathbf{\bar{f}}_i^{n+1} - \mathbf{\bar{f}}_i^n}{\Delta t} = \mathbf{L}(\mathbf{\bar{f}}_i^n)$$

(44)

is applied to solve the system of ordinary differential equations (37). As opposed to implicit time integration schemes, we avoid solving for the nonlinearity by applying an explicit approach

$$\mathbf{M} \mathbf{\Delta} \mathbf{\bar{f}}_i^{n+1} = \Delta t \mathbf{L}(\mathbf{\bar{f}}_i^n), \quad i = 0, 1, 2, \ldots, 8$$

(45)

$$\mathbf{\bar{f}}_i^{n+1} = \mathbf{\bar{f}}_i^n + \mathbf{\Delta} \mathbf{\bar{f}}_i^{n+1}$$

(46)

where the structure of the coefficient or mass matrix $\mathbf{M}$ is as follows:

$$
\begin{bmatrix}
M_0 & 0 & 0 & 0 & \cdots & 0 \\
0 & M_1 & 0 & 0 & \cdots & 0 \\
0 & 0 & M_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & M_8
\end{bmatrix}
\begin{bmatrix}
\mathbf{\Delta} \mathbf{\bar{f}}_0^{n+1} \\
\mathbf{\Delta} \mathbf{\bar{f}}_1^{n+1} \\
\mathbf{\Delta} \mathbf{\bar{f}}_2^{n+1} \\
\vdots \\
\mathbf{\Delta} \mathbf{\bar{f}}_8^{n+1}
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{L}(\mathbf{\bar{f}}_0^n) \\
\mathbf{L}(\mathbf{\bar{f}}_1^n) \\
\mathbf{L}(\mathbf{\bar{f}}_2^n) \\
\vdots \\
\mathbf{L}(\mathbf{\bar{f}}_8^n)
\end{bmatrix}
$$

(47)
Note that the submatrices \( M_i, i = 0, 1, 2, \ldots, 8 \) are identical, decoupled and independent of the solution vector \( \hat{f}_i^{n+1} \). Therefore, we have for each time step \( n \) a linear system which can be solved by performing an LU decomposition only once. Since the submatrices \( M_i, i = 0, 1, 2, \ldots, 8 \) are identical, it is sufficient to determine the inverse of one submatrix. During the time stepping, one has to compute the right-hand side and to perform a forward- and backsubstitution in order to find the new increment \( \Delta f_i^{n+1} \). Having found the solution for time step \( n + 1 \), the macroscopic quantities like the density, velocity and pressure are computed by

\[
\hat{f}_i^{n+1} = \sum_{k=1}^{k_{\text{modes}}} N_k f_k^{n+1} = N f_i^{n+1} 
\]

(48)

\[
\rho^{n+1} = \sum_{i=0}^{8} \hat{f}_i^{n+1} 
\]

(49)

\[
\mathbf{u}^{n+1} = \frac{1}{\rho^{n+1}} \sum_{i=0}^{8} \hat{f}_i^{n+1} \mathbf{e}_i 
\]

(50)

\[
\hat{p}^{n+1} = c_s^2 \rho^{n+1} 
\]

(51)

Remarks

(i) Equation system (47) corresponds to the discretization with one element only. If more elements are used, the structure of the equation system remains block diagonal since the discontinuous Galerkin approach is applied. The coupling of elements is achieved by means of the right-hand side whereas the element mass matrices are decoupled.

(ii) Although an explicit time integration scheme is applied, a linear equation system has to be solved. If one had used a basis which yields diagonal mass matrices or applied mass lumping, the solution of the equation system would be trivial.

(iii) Compared to the high-order finite element approach, a first-order forward Euler scheme for the temporal discretization seems to be somehow 'unbalanced'. Instead of applying a forward Euler scheme, one could use any other time integration scheme of higher order. In recent years, there has been ongoing research on the development of Runge Kutta discontinuous Galerkin (RKDG) methods for non-linear hyperbolic systems [31–33]. The results of this work could be applied to the (temporal) discretization of the discrete Boltzmann equation.

(iv) As mentioned in Section 3.1, the initial conditions are computed by performing an \( L_2 \)-projection of the equilibrium distribution function onto the finite element space. This procedure calls for assembling and inverting the mass matrix. Since this mass matrix is the same as the one in (44), the assembly and LU decomposition can be already performed during the computation of the initial conditions and may be re-used during the time stepping scheme.
4. NUMERICAL EXAMPLES

Four numerical examples are presented in this section and the results obtained with the proposed discontinuous Galerkin scheme are compared to the exact NS solution. The main objective is to investigate the performance of high-order finite elements. Since the selected examples for which analytic NS solution are available are quite simple, one element is sufficient to discretize the computational domain. Therefore, the difference between a continuous and a discontinuous Galerkin approach is defined in the way of how the boundary conditions are treated. Unless otherwise stated, the initial conditions correspond to zero velocity, $u = 0$, and the initial density is chosen as $\rho = 1$.

4.1. Couette flow

As a first example the shear flow between two parallel plates is computed. The geometry and boundary conditions are sketched in Figure 6. The horizontal velocity of the upper plate is $u_w = 1$ while the lower plate is at rest. $L = W = 32$ denote the length and the width of the channel, respectively. $v = \nu/3 = 1/3$ corresponds to the kinematic viscosity, and the density of the fluid is $\rho = 1$. The exact NS solution $u_{NS} = (u_x, u_y)^T$ for the stationary problem reads:

$$
\begin{align*}
    u_x(y) &= u_w \frac{y}{W} \\
    u_y &= 0
\end{align*}
$$

The computational domain is discretized with one element of degree $p = 1$ and the time step size is chosen to be $\Delta t = 1.5$. Periodic boundary conditions are applied on the left- and the right-hand side of the channel. At the upper and the lower plates both types of boundary condition are tested, the Inamuro’s as well as the Ladd’s method. The computation is continued until a stationary solution is found. For both types of boundary conditions, the approximation of the

![Figure 6. Couette flow.](https://example.com/figure6)

D2Q9-DG element with $p = 1$ converges to the exact solution down to machine accuracy. The error of the velocity $\| \mathbf{u}_{NS} - \mathbf{u}_{D2Q9-DG} \|_{L^2}$ in $L_2$-norm amounts to approximately $1.0 \times 10^{-14}$.

4.2. Plane Poiseuille flow

As a second example a forced channel flow is computed (Figure 7). The length and the width of the channel are $L = W = 32$, the density of the fluid is $\rho = 1$ and the kinematic viscosity amounts to $\nu = \tau/3 = 1/3$. A body force acting in the horizontal direction $F_x = 2.60416E - 05$ is accelerating the fluid. Again, periodic boundary conditions are applied on the left- and the right-hand sides of the channel. Both types of boundary conditions are applied, Ladd’s as well as Inamuro’s method.

The exact NS solution $\mathbf{u}_{NS} = (u_x, u_y)^T$ for the stationary problem reads:

$$
\begin{align*}
  u_x &= u_{\max} \left( 1 - \left( \frac{2y}{W} - 1 \right)^2 \right) \\
  u_y &= 0
\end{align*}
$$

(53)

where $u_{\max} = F_x W^2 / (8 \rho \nu) = 0.01$. The channel is discretized with one element of order $p = 2$ and a time step size of $\Delta t = 1$ has been chosen to find a stationary solution. After approximately 10 000 time steps, the solution becomes stationary. The error of the velocity in $L_2$-norm amounts to approximately $8.3 \times 10^{-4}$ when Ladd’s boundary conditions are applied and $1.1 \times 10^{-13}$ in the case of Inamuro’s method. For both approaches the velocity $u_x(x = 16, y)$ computed along a vertical line at $x = 16$ and $y \in [0, 32]$ are compared to the exact NS solution in Figure 8. Since in Figure 8 almost no difference between the different solutions can be seen, the error $|u_{x,NS} - u_{x,D2Q9-DG}|(x = 16, y)$ is plotted in Figure 9 on a logarithmic scale. From this, it is obvious that Ladd’s boundary condition introduces a slip velocity which deteriorates the accuracy of the approximation. However, applying Inamuro’s method reduces the error down to machine accuracy.

![Figure 7. Poiseuille flow.](image)

Figure 8. Comparison of the NS solution with the D2Q9-DG approximation.

Figure 9. Comparison of the NS solution with the D2Q9-DG approximation.

So far, the computation of the forced channel flow has been carried out for a low Reynolds number

\[ Re = \frac{u_{\text{max}} W}{\nu} = \frac{9F_x W^3}{8\rho\tau^2} \]

Figure 10. Maximum time step size for stable simulations with different Reynolds numbers.

with $Re = 0.96$. In general, flows at high Reynolds numbers are more interesting. To increase the Reynolds number the kinematic viscosity $v = \tau/3$ is successively decreased. The channel flow is recomputed for a wide range of Reynolds numbers with a polynomial degree of $p = 2$ applying Inamuro's method to define the boundary conditions at the wall. As the Reynolds number is increased, the time step size $\Delta t$ has to be reduced. In Figure 10, the maximum time step size still yielding a stable (temporal) discretization is plotted against the Reynolds number. From this it is obvious that the proposed method can be applied also for the computation of flows at higher Reynolds numbers. However, it is clear that a simple forward Euler time integration should be replaced by a more advanced method increasing the overall efficiency.

Remark

For Reynolds number $Re > 3000$ the flow turns from laminar to turbulent which mathematically manifests itself in the existence of multiple solutions. In the reported experiment, the low spatial resolution (a single biquadratic element) precluded development of more complex approximate solutions.

4.3. Rotating Couette flow

The third example to be considered has been investigated by Peng et al. [14] in order to demonstrate the flexibility of a finite volume scheme for the discrete Boltzmann equation. The shear flow between two coaxial cylinders, see Figure 11, is computed and compared to the exact NS solution

$$u_\phi(r) = \frac{1}{R_o^2 - R_i^2} \left( (\Omega_o R_o^2 - \Omega_i R_i^2)r + (\Omega_i - \Omega_o) \frac{R_i^2 R_o^2}{r} \right)$$

(55)
The radii are $R_i = 30$, $R_o = 60$ and the angular velocities are set to $\Omega_i = 0$ and $\Omega_o = 1/(100R_o)$. The density of the fluid is $\rho = 1$ and the kinematic viscosity $\nu = \tau/3$ is varied such that $\tau = [0.5, 1.0, 1.5, 2.0]$.

Half of the domain is discretized with only one element, taking advantage of the blending function method which allows to exactly describe the geometry of the problem. At the curved boundaries, see Figure 12, Ladd's boundary condition has been applied to enforce the angular velocities whereas at the horizontal lines periodic boundary conditions have been selected to reduce the size of the computational domain. The computations are performed with a polynomial degree of $p = 8$ and a time step size of $\Delta t = 0.25$. In order to neglect the error of the numerical integration of the element matrices and vectors, a Gaussian scheme with $(p + 8)^2$ integration points has been applied. A contour plot of the velocity for $\tau = 0.5$ is plotted in Figure 12.

From Equation (55) it is obvious that the exact NS solution is independent of the kinematic viscosity $\nu = \tau/3$. Therefore, it is interesting to investigate the influence of the relaxation time $\tau$ with respect to the D2Q9-DG approximation by comparing the results to the exact NS solution. In Figure 13, the error of the velocity $|u_{NS} - u_{D2Q9-DG}|(x = 0, y)$ is plotted for $\tau = 2.0$ and $0.5$ along a cutline $x = 0$, $y \in [R_i, R_o]$. From this it is obvious that the deviation of the solution of the D2Q9 model from the NS solution is reduced as $\tau$ is decreased as expected from the results of the Chapman-Enskog analysis.

In order to investigate the influence of the relaxation time $\tau$ in more detail, the relative error in $L_2$-norm $\|u_{NS} - u_{D2Q9-DG}\|_{L_2}/\|u_{NS}\|_{L_2}$ against the number of degrees of freedom for a $p$-extension with $p = 1, \ldots, 8$, is plotted in Figure 14. It is obvious from Figure 14 that the error reduces as $\tau$ is decreased. It can also be seen that, beyond a critical value, a further increase in polynomial degree does not improve the accuracy of the approximation. The remaining error can be interpreted as a modelling error, i.e. the deviation of the D2Q9 model from the NS equation. Note that the Knudsen number $\varepsilon = (c_r/L_r)\tau$, where $c_r$ is a characteristic microscopic velocity and $L_r$ is a macroscopic reference length is proportional to the relaxation time $\tau$ [45]. It is known from the literature that the hydrodynamic (continuous) regime is characterized by
small Knudsen numbers [2]. Therefore, the approximation of the exact solution of the D2Q9 model converges to NS solution as the relaxation time $\tau$ is reduced.

In Figure 15, the convergence of the relative error of D2Q9-DG approximation with respect to the exact NS solution is plotted against the relaxation time $\tau$ in a double logarithmic style.

Copyright © 2006 John Wiley & Sons, Ltd.  
Figure 14. Comparison of the NS solution with the D2Q9-DG approximation.

From this, it is evident that the error is of order $O(\varepsilon^n)$ like it is expected from the literature. It has to be mentioned that in addition to a small Knudsen number, the Mach number has also to be small in order to recover the incompressible NS solution [45].

4.4. Taylor vortex flow

As the last example, the (instationary) Taylor vortex flow is considered, see Figure 16.

The corresponding NS solution reads

$$u_x(x, y, t) = -u_0 \exp(-vt(k_1^2 + k_2^2)) \cos(k_1x) \sin(k_2y)$$

$$u_y(x, y, t) = u_0 k_1/k_2 \exp(-vt(k_1^2 + k_2^2)) \sin(k_1x) \cos(k_2y)$$

(56)

where $u_0 = 0.01$, $k_1 = 2\pi/L$ and $k_2 = 2\pi/W$ with $L = W = 32$ denoting the length and width of the domain. The kinematic viscosity of the fluid is $\nu = \tau/3 = 1/6$, and the density is set to $\rho = 1$. The initial conditions (29) are computed by inserting velocity (56) and density $\rho = 1$ at $t = 0$ into the equilibrium distribution function, and performing the $L_2$-projection into the finite element space. Again, the whole domain is discretized with one element and the time step size is chosen to be $\Delta t = 0.01$. Periodic boundary conditions are applied in $x$ as well as in $y$ direction. In absence of driving forces, the velocity will decay exponentially. This example has been investigated by Peng et al. [15] for a finite volume scheme for the discrete Boltzmann equation.

A comparison of the NS solution and the D2Q9-DG approximation with $p = 4$ and $8$ is presented in Figure 17, for different time levels $t = [0, 22, 54, 108, 600]$. The velocity component $u_y(x = 12, y, t)$ is plotted along a cutline $x = 12$, $y \in [0, 12]$. From this, it is evident that a
Figure 15. Influence of the relaxation time $\tau$.

Figure 16. Taylor vortex flow. Left-hand side, $u_x(x, y, t = 0)$; right-hand side, $u_y(x, y, t = 0)$.

degree of $p = 4$ shows significant deviations from the NS solution while $p = 8$ yields an accurate approximation.

To investigate the accuracy of the D2Q9-DG approach in more detail, the relative error in $L_2$-norm with respect to the NS solution is plotted in Figure 18 against the number of degrees of freedom for an increase in polynomial degree $p = 2, \ldots, 9$ at time $t = 0$ and 54. The curve for $t = 0$ presents the accuracy of the $L_2$-projection of the initial conditions. An exponential convergence can be clearly observed. For $t = 54$, three different curves are presented, where the number of Gaussian points $GO^2$ is varied and the time step size is also changed. From this curves it is obvious that beyond $p = 7$ an increase in polynomial degree does not improve the
accuracy further. Raising the number of integration points or choosing a smaller time step size has also no significant influence on the accuracy of the D2Q9-DG approximation. From this, it is evident that the exponential convergence of the p-extension is limited by the modelling error, i.e. the difference between the exact solution of the D2Q9 model, and the NS solution. Despite the modelling error, an approximation with less than 0.5% error in the $L_2$-norm can be obtained with only one element and $p = 7$.

To further investigate the modelling error, we recompute the same problem for different values of $\tau = \{0.03125, 0.0625, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0\}$ where $u_0$ has been chosen such
that the Reynolds number

$$Re = \frac{u_0 L}{v} = \frac{3u_0 L}{\tau} = 1.92$$ (57)

remains the same value as in the previous runs. The computations have been carried out for different polynomial degrees $p$ with a constant time step size $\Delta t = 0.01$ and $(p + 3)^2$ integration points. In Figure 19, the relative error of the velocity in $L_2$-norm is reported. The time $t$ at which the D2Q9-DG approximation has been compared to the exact NS solution is scaled by the relaxation time $\tau t = 27$. Therefore, the solution at time $t$ remains similar for the different values of $\tau$. From Figure 19, it is evident that the modelling error related to the Knudsen number $\varepsilon = (c_s/L) \tau$ as well as the Mach number $Ma = u_r/c_s$ decreases with second order as $\tau$ and $u_r = u_0$ are reduced. Furthermore, it is interesting to note that odd polynomial degrees improve the quality of the D2Q9-DG approximation more efficiently than degrees of even order which is due to the symmetry behaviour of the exact solution.

So far, the computation of the Taylor vortex flow has been carried out for a low Reynolds number (57). Thinking of more realistic problems it is of course important to be able to simulate fluid flows with higher Reynolds numbers. To address this point, the Taylor vortex flow is computed at different Reynolds numbers. Since the flow speed has to be small compared to the isothermal speed of sound $c_s$, the value of $u_0$ is kept fixed at 0.01 and the kinematic viscosity $v = c_s^2 \tau = \tau / 3$ is varied instead. The simulations are carried out with a polynomial degree of $p = 8$ and $(p + 1)^2$ integration points for different values of $\tau = \{0.96, 0.096, 0.0096\}$ yielding the following Reynolds numbers $Re = \{1, 10, 100\}$. The time step size $\Delta t$ has been set to 0.01. No attempt has been made to find an optimal value of $\Delta t$ which would have reduced the corresponding number of time steps.
Figure 19. Influence of the modelling error based on the Knudsen number $\varepsilon$ and the Mach number $Ma$.

Figure 20. Relative error of the D2Q9-DG simulation for different Reynolds numbers.

In Figure 20, the relative error of the velocity in $L_2$-norm for different Reynolds numbers is plotted against the time $t$. The simulations have been carried out for a wide range of $t$. Since velocity (56) decays exponentially as $t$ is increased, the error $\|u_{NS}(t) - u_{D2Q9-DG}(t)\|_{L_2}$ is...
related to the initial value $\|u_{NS}(t=0)\|_{L^2}$ of the exact NS solution in order to prevent dividing the absolute error by zero. From Figure 20, it can be seen that the error of all computations is nowhere above 2.5%. This demonstrates that D2Q9-DG approach may also be applied to more demanding simulations with higher Reynolds numbers. Since an explicit scheme is applied, the number of time steps depends on the eigenspectrum and time-step restriction of the weak advection operator. Therefore, future work will focus also on an improved combination of the finite element basis and an appropriate time stepping scheme as discussed for the linear advection equation, e.g. in References [31,42].

5. CONCLUSIONS

A discontinuous Galerkin formulation of high order for the discrete Boltzmann equation, based on the D2Q9 model has been presented. Different kinds of boundary conditions have been considered by investigating four examples for which exact Navier–Stokes solutions are available. It is numerically shown that the proposed D2Q9-DG formulation allows to obtain the exact NS solution of a shear flow between two parallel plates (Couette flow) as well as for a forced channel flow (plane Poiseuille flow). Considering the shear flow between two coaxial cylinders (rotating Couette flow), half of the computational domain is discretized with only one curved element taking advantage of the blending function method. A $p$-extension yields a pre-asymptotic exponential rate of convergence of the error in velocity in $L_2$-norm which is limited by the modelling error. The modelling error, i.e. the deviation of the exact solution of the D2Q9 model and the NS equation depends on the Mach number as well as Knudsen number as expected from the Chapman–Enskog analysis. For a fixed Reynolds number it is demonstrated that the modelling error can be reduced by adjusting the Mach number and the kinematic viscosity. Therefore, very accurate approximations can be efficiently obtained due to an exponential rate of convergence, even for instationary problems.

Future work will enhance this concept to unstructured meshes in two as well as three dimensions. Furthermore, it is planned to apply integration methods of Runge–Kutta type which can be extended to adaptive time stepping schemes. Considering high-order finite element schemes, appropriate meshing is a very important issue. The mesh generator should provide coarse meshes which are locally strongly refined ($hp$-type meshes) towards points or lines of singularities. Furthermore, it is important to accurately account for the exact geometry of the problem. Therefore, the mesh generator should provide additional information about the geometry at element edges and element faces (in 3D). For 2D problems $hp$-meshing strategies and generators are discussed, for example, in References [46–50]. In 3D it is planned to apply a discontinuous tetrahedral element formulation based on the D3Q15 model [27]. For tetrahedral elements $hp$-mesh generators are available, see References [51–55]. It is expected that the main problem in 3D will be the efficient implementation of a tetrahedral finite element in order to fully exploit the advantage of high-order methods. In order to efficiently compute accurate approximations, adaptivity in space and time will be mandatory. This will increase the complexity of the problem dramatically, especially when applying $hp$-refinement where one has to decide whether to refine in $h$ or in $p$. However, it was shown by different authors for several model problems in solid as well as fluid mechanics and also for electromagnetics [37,56–63] that $hp$-extensions are by far the most efficient methods and in many cases the only approach to obtain an approximation with a prescribed accuracy.
ACKNOWLEDGEMENTS

The first author would like to thank Prof. Manfred Krafczyk and Dr Jonas Tölke for fruitful discussion about the discrete Boltzmann equations and for encouraging him to apply high-order finite elements to this problem. Furthermore, we would like to thank Prof. Irene M. Gamba for a vivid discussion giving insight into Boltzmann-type equations. Part of the work of the first author has been supported by the J.T. Oden Research Faculty fellowship. This support is gratefully acknowledged.

REFERENCES


