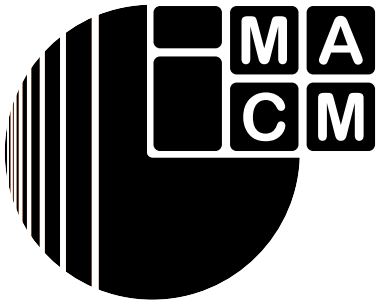


Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 15/04
revised work of preprint 14/20

Daniel Heubes, Andreas Bartel, Matthias Ehrhardt

Concept for a one-dimensional discrete artificial boundary condition for the lattice Boltzmann method

January 14, 2015

http://www.math.uni-wuppertal.de

Concept for a one-dimensional discrete artificial boundary condition for the lattice Boltzmann method

Daniel Heubes*, Andreas Bartel, Matthias Ehrhardt

Bergische Universität Wuppertal, Fachbereich Mathematik und Naturwissenschaften, Lehrstuhl für Angewandte Mathematik und Numerische Analysis, Gaußstraße 20, 42119 Wuppertal, Germany

Abstract

This article deals with artificial boundaries which you encounter when a large spatial domain is confined to a smaller computational domain. Such an artificial boundary condition should not preferably interact with the fluid at all. Standard boundary conditions, e.g., a pressure or velocity condition, result in unphysical reflections. So far, existing artificial boundary conditions for the lattice Boltzmann method (LBM) are transferred from macroscopic formulations.
In this work we propose novel discrete artificial boundary conditions (ABCs) which are tailored on the LBM's mesoscopic level. They are derived directly for the chosen LBM with the aim of higher accuracy. We describe the idea of discrete ABCs in a three velocity (D1Q3) model governing the Navier-Stokes equations in one dimension. We emphasize the fact that our approach has the approved potential to be generalized to higher dimensions and general collision models. Numerical results finally demonstrate the superiority of our new boundary condition in terms of accuracy compared to previously used ABCs.

Keywords: lattice Boltzmann method, artificial boundary conditions, unbounded domain, D1Q3, characteristic boundary conditions, impedance boundary condition

1. Introduction

The *lattice Boltzmann method* (LBM) is an established numerical method of *computational fluid dynamics* (CFD) [1, 2, 3]. In this theoretical work we consider

*corresponding author
Email addresses: heubes@math.uni-wuppertal.de (Daniel Heubes), bartel@math.uni-wuppertal.de (Andreas Bartel), ehrhardt@math.uni-wuppertal.de (Matthias Ehrhardt)

the method only in one spatial dimension. In higher dimensions the LBM has shown to be attractive for real-world simulations not only for its ease of implementation but also for its applicability to complex flows. Examples of applications can be found, e.g., in [4, 5, 6]. Nevertheless, this theoretical work in one spatial dimension is the inevitable first step to develop our new strategies, which we will transfer to higher space dimensions afterwards.

In simulations often non-physical boundaries occur, e.g., if a large fluid domain is confined to a smaller computational domain. Like all boundaries, also these non-physical boundaries, so-called *artificial boundaries*, need boundary conditions in a numerical implementation. Ideally, they should be chosen such that they do not interact with the fluid in an unwanted way, i.e., that no spurious effects influence the simulation results. Boundary conditions for the LBM are usually derived from known macroscopic physical conditions. Anyway, the problem of the correct boundary condition for artificial boundaries is also given on the macroscopic scale.

On the macroscopic scale, several studies on *artificial boundary conditions* (ABCs) in CFD were performed. The pioneering work for absorbing boundary conditions for wave equations was established by Engquist and Majda [7]. Hedstrom [8] and Thompson [9] developed non-reflecting *characteristic boundary conditions* (CBCs) in the field of nonlinear hyperbolic equations. Kröner [10] derived approximate exact absorbing boundary conditions for the two-dimensional linear Euler equations. Non-reflecting boundary conditions (NRBCs) for the Navier-Stokes equations were presented by Poinsot and Lele [11]. A review on absorbing boundary conditions for hyperbolic systems can be found in [12].

Artificial boundary conditions for the lattice Boltzmann method are less frequently studied. A macroscopic formulation (i.e., PDE-based) of an ABC cannot be applied directly. If at all possible, it has to be transferred to the mesoscopic level of the LBM. Such a transfer was done by Izquierdo and Fueyo [13] and Kim et al. [14]. These authors formulated non-reflecting boundary conditions for the LBM by adapting macroscopic CBCs. In these articles, an analysis of the characteristics of a system for macroscopic quantities is performed to construct a system with reduced unphysical reflections.

Another approach of an ABC for the LBM was pursued recently by Schlaffer [15]. He formulated a so-called impedance boundary condition, in which the acoustic impedance of the fluid is chosen appropriately.

Furthermore, Najafi-Yazdi and Mongeau [16] as well as Tekitek et al. [17] developed an absorbing layer technique for the LBM which is based on the *perfectly matched layer* (PML) concept of Bérenger [18]. To be more precisely, the PML technique is not a boundary condition as such. It introduces a damping zone in

the computational domain, a boundary condition is still required. Best results with damping zones are expected in hybrid approaches, when the PML technique is combined with ABCs.

In an earlier work [19], we derived a first exact ABC for the D1Q2 LBM on the discrete level. It is based on a linear collision operator, which results in an advection problem on the macroscopic scale. The main goal of the current work is to extend the ABC [19] to non-linear collision operators. To this end, we only consider the D1Q3 model, that is the simplest lattice Boltzmann model capable of simulating 1D-Navier-Stokes flows. As a further objective, this theoretical one-dimensional work will be the basis for future works on discrete ABCs, including those in more space dimensions [20]. Thus, for the first time, this work presents a one-dimensional ABC for the (Navier-Stokes) LBM which is constructed purely on the discrete level. In fact, our tree representation is applicable to any non-linear collision operator and thus is some sense an universal approach.

This article is structured as follows. In Section 2 we give a short recapitulation of the LBM, in which we focus on the three velocity model in 1D (D1Q3). The tree interpretation for a description of the evolution of populations introduced in [19] is generalized for arbitrary collision models in Section 3. Section 4 states the present boundary condition. Afterwards in Section 5 we explain an efficient implementation of it and also enhance the basic algorithm to improve the efficiency. This paper ends with a presentation of numerical tests in Section 6 and conclusions in Section 7.

2. The D1Q3 lattice Boltzmann method

We consider the LBM in one spatial dimension with discrete velocities $c_0 = 0$, $c_1 = -1$ and $c_2 = 1$ (D1Q3). Given an equidistant grid (lattice), where space and time points are referred to as x_n and t_s, respectively. The so-called *populations* f_i give the number density of fictitious particles with velocity c_i at each lattice node (x_n, t_s). The notation $f_i(n, s) := f_i(x_n, t_s)$ is used as abbreviation for the populations at the lattice node (x_n, t_s). For each node a vector can be created by grouping all populations:

$$\vec{f}(n, s) = \big(f_0(n, s),\ f_1(n, s),\ f_2(n, s)\big)^\top.$$

Now, the *lattice Boltzmann equation* (LBE) defines an update rule of the populations via an explicit formula:

$$f_i(n + c_i, s + 1) = f_i(n, s) + C_i\big(\vec{f}(n, s)\big), \quad \text{for} \quad i = 0, 1, 2. \tag{1}$$

The right-hand side term $C_i(\vec{f})$ denotes the collision operator, which models the local particle interaction. Here we use the popular single time BGK collision model (due

to Bhatnagar, Gross and Krook [21])

$$C_i\big(\vec{f}(n,s)\big) = -\omega\big(f_i(n,s) - E_i(n,s)\big), \tag{2}$$

with relaxation parameter $\tau = \frac{1}{\omega}$. The local equilibrium distribution E_i ($i \in \{0,1,2\}$) with weights $w_0 = \frac{2}{3}$ and $w_1 = w_2 = \frac{1}{6}$ reads:

$$E_i\big(\rho(n,s), u(n,s)\big) := w_i\rho\left[1 + 3c_iu + \frac{9}{2}(c_iu)^2 - \frac{3}{2}u^2\right]. \tag{3}$$

E_i is chosen such that the mass density

$$\rho = \rho(n,s) = f_0(n,s) + f_1(n,s) + f_2(n,s)$$

and the fluid velocity

$$u = u(n,s) = \frac{1}{\rho(n,s)}\Big(f_2(n,s) - f_1(n,s)\Big),$$

evolve according to the Navier-Stokes equations [22].

Instead of using the BGK approximation (2), another popular choice for the collision operator $C_i(\vec{f})$ is given by a multi relaxation time model. By this model also the Navier-Stokes equations can be recovered. Also other macroscopic evolution equations than the Navier-Stokes equations can be realized. In the BGK approximation this is done by substituting the equilibrium distribution (3). By this, e.g., an advection-diffusion equation can be realized. What have all collision models in common, is that the LBE (1) consists of two basic steps:

- collision – the evaluation of the right-hand side,
- transport – time propagation / assignment to left-hand side.

With the above introduced collision operator (2)–(3) the complete collision process can be described with non-linear functions in terms of populations:

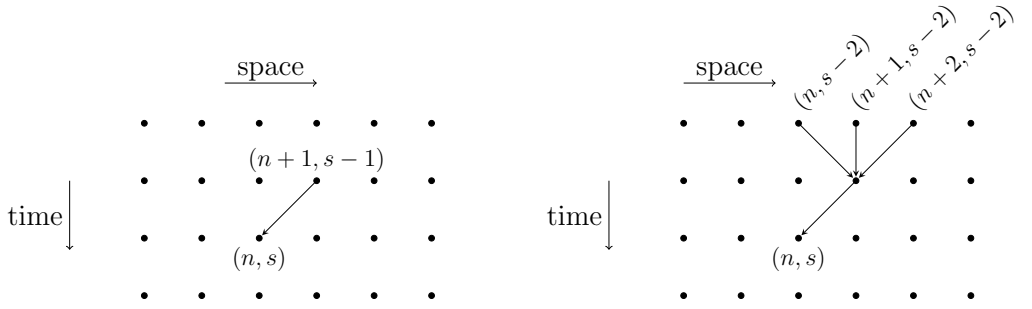
$$g_0(\vec{f}) = g_0(f_0,f_1,f_2) = f_0 - \frac{\omega}{3}\frac{{f_0}^2 + {f_1}^2 + {f_2}^2 - f_0(f_1+f_2) - 10f_1f_2}{f_0+f_1+f_2},$$
$$g_{1,2}(\vec{f}) = g_{1,2}(f_0,f_1,f_2) = f_{1,2} + \frac{\omega}{6}\frac{{f_0}^2 + {f_1}^2 + {f_2}^2 - f_0(f_1+f_2) - 10f_1f_2}{f_0+f_1+f_2}.$$

Here the g_i's are post-collision populations. In the upcoming section the g_i's can be any post-collision populations, that means there is no restriction to the Navier-Stokes collision operator (2)–(3). I.e., they are evaluations of the right hand side terms of the LBE (1).

$$g_i(\vec{f}) = f_i + C_i(\vec{f}). \tag{4}$$

3. The evolution of populations

To get a better understanding of the populations, we consider space-time diagrams to track their evolution. Without loss of generality we focus on the computation of an f_1 population at an arbitrary lattice node (n, s). The deterministic LBE (1) expresses the pre-collision population $f_1(n, s)$ as a function of the three pre-collision populations from node $(n + 1, s - 1)$. This is visualized in Fig. 1(a), where the directed edge symbolizes the transport of the post-collision population $g_1\big(\vec{f}(n + 1, s - 1)\big)$. The source node of a directed edge shall always contain a local



(a) Computation in terms of $f_i(n + 1, s - 1)$.

(b) Computation in terms of $f_i(m, s - 2)$ with $m \in \{n, n + 1, n + 2\}$.

Figure 1: Space-time diagrams visualizing the computation of a population (here $f_1(n, s)$) in terms of populations of past nodes.

collision according to (4), i.e., it holds:

$$f_1(n, s) = g_1\big(f_0(n + 1, s - 1), f_1(n + 1, s - 1), f_2(n + 1, s - 1)\big). \tag{5}$$

This formulation assumes that the populations at node $(n + 1, s - 1)$ are known.

Now, we assume that only populations up to time level $t = s - 2$ are known. Then each $f_i(n+1, s-1)$ has to be represented in terms of pre-collision populations $f_i(m, s - 2)$, $m \in \{n, n + 1, n + 2\}$. We get:

$$f_1(n, s) = g_1\Big(g_0\big(\vec{f}(n + 1, s - 2)\big), g_1\big(\vec{f}(n + 2, s - 2)\big), g_2\big(\vec{f}(n, s - 2)\big)\Big).$$

The corresponding space-time diagram is shown in Fig. 1(b). If, as a last exemplary case, the population $f_0(n + 1, s - 1)$ is known, but neither $f_1(n + 1, s - 1)$ nor $f_2(n + 1, s - 1)$, we obtain

$$f_1(n, s) = g_1\Big(f_0(n + 1, s - 1), g_1\big(\vec{f}(n + 2, s - 2)\big), g_2\big(\vec{f}(n, s - 2)\big)\Big)$$

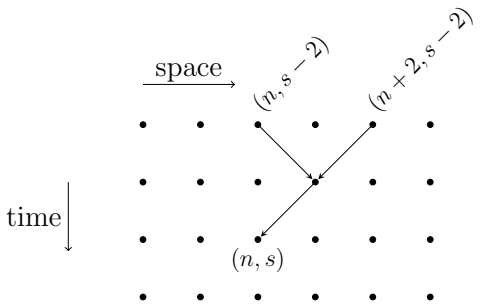


Figure 2: Space-time diagram analog to the previous figure. Here the population $f_1(n, s)$ is visualized in terms of $f_0(n+1, s-1)$ and $f_i(m, s-2)$, for $m \in \{n,\, n+2\}$ and $i \in \{0,\, 1,\, 2\}$.

and Fig. 2 as the corresponding diagram.

To sum up, we can read our space-time graphs as follows: A directed edge from $(n{+}1,\ s{-}1)$ to $(n,\ s)$ (south-western direction) represents the pre-collision population $f_1(n,\ s)$ in the destination node. In fact, that edge represents the population f_1 in terms of the pre-collision populations at the source of the edge $(n+1,\ s-1)$ as stated by (5). Similarly, an edge of southern direction represents the f_0 and an edge in south-eastern direction represents the f_2 population in the destination node. By induction, we can traverse any graph several layers into the past. To this end, at each node three populations have to be used in the collision. Now, a missing incoming edge signifies that the corresponding population is known at the corresponding node. Thus, given a directed graph in a space-time diagram, the observations above are sufficient to state the calculation rules.

4. Construction of a discrete artificial boundary condition

Without loss of generality we consider a right boundary of the computational domain to formulate a *discrete artificial boundary condition* (DABC) for the D1Q3 lattice Boltzmann model. The right boundary shall be located at the node with position $x = x_N$. A (right) boundary condition has to provide the inward populations $f_1(N, s)$ for all time levels $s \in \mathbb{N}^+$. To compensate the lack of right neighbors, we assume to have fictitious nodes in the exterior domain $x > x_N$. These nodes shall be initialized by known populations, e.g., equilibrium distributions, which are computed from a suitable and given density ρ_{ext} and velocity $u_{\text{ext}} \in \mathbb{R}$. I.e., we assume

$$f_i(m, 0) = E_i^{\text{ext}} := E_i(\rho_{\text{ext}}, u_{\text{ext}}), \quad \text{for} \quad i = 0,\, 1,\, 2, \quad \text{and} \quad m > N. \tag{6}$$

For simplicity we assume (6) holds in the sequel.

Theoretically, we can formulate an exact DABC based on the given assumptions, where one has to consider a set of directed edges going back to the initial layer like shown in the space-time diagram of Fig. 3.

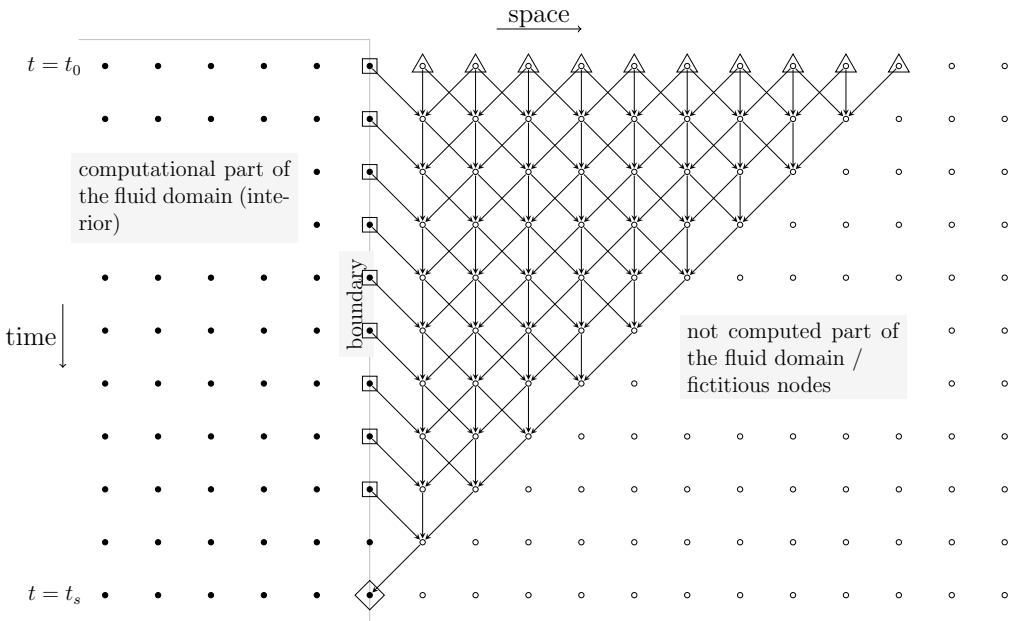


Figure 3: This space-time diagram explains the basic idea of an exact DABC. In the exterior of the computational domain some fictitious nodes are introduced (marked by circles). Interior nodes of the computational domain are denoted by filled circles. The unknown population is expressed in terms of initial information (triangles) and past boundary information (squares).

The exact DABC for the inward population $f_1(N, s)$ (diamond marked node) depends on populations of two types of nodes:

(i) boundary nodes at previous time levels (square marked nodes): $f_i(N, \ell)$ with $0 \leq \ell < s - 2$ (excluding the time level $s - 1$) as well as
(ii) exterior (fictitious) nodes (marked by triangles): E_i^{ext}.

In both cases holds $i \in \{0, 1, 2\}$, since each edge includes a collision at its source node, see Section 3. In other words, the usual lattice Boltzmann approach is also applied for the fictitious nodes in the exterior domain, cf. Fig. 3.

However, this formulation is not efficient at all, since the required effort increases rapidly with the number of time steps simulated. For a more efficient approximation of the inward populations $f_1(N, \cdot)$, we propose to incorporate boundary populations of at most $H_{\max} \in \mathbb{N}$ past levels. The number $H_{\max}$ is called the *maximal history*

depth of the approximate DABC. More precisely, in order to compute the inward population $f_1(N, s)$, we incorporate the boundary populations $f_i(N, \ell)$, with $s - H(s) \leq \ell \leq s - 2$, see Fig. 4. Furthermore, $f_1(N, s)$ may still depend on E_i^{ext}. The

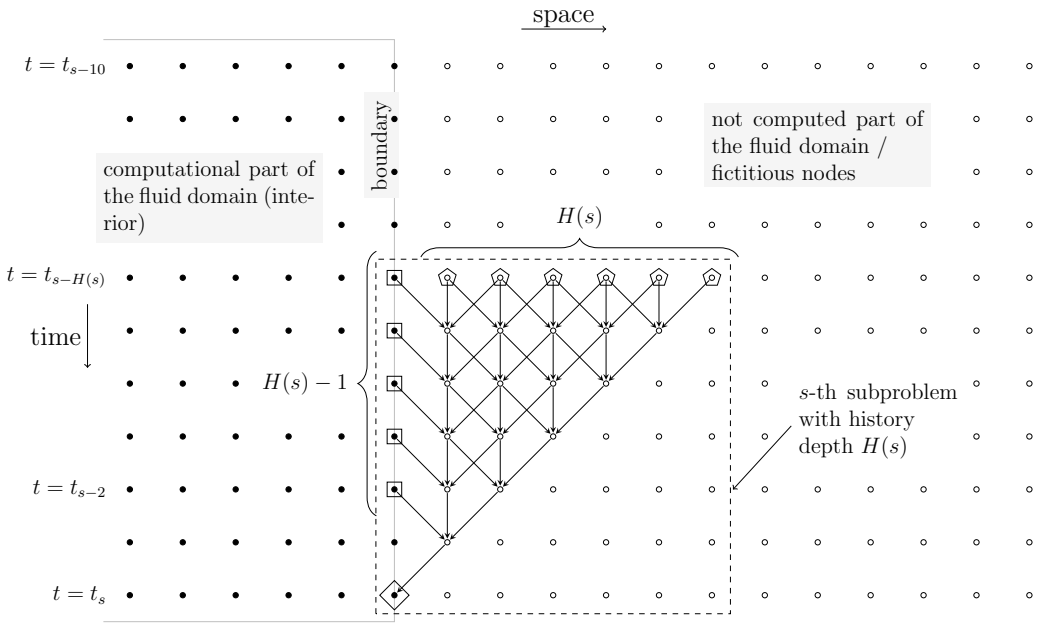


Figure 4: This space-time diagram visualizes the approach of (approximate) DABC. The unknown population is expressed in terms of past boundary information (squares) and some fictitious nodes (pentagons). A truncation parameter H (history depth) controls the quantity of nodes entering the computation.

number $H(s)$ (with $H(s) \leq H_{\max}$) is the history depth at time level s. Thus, as a function of the time level s the history depth is allowed to vary during the simulation. The history depth $H(s)$ must not exceed the time level s, i.e., we require $H(s) \leq s$. The above exact boundary condition is achieved by adapting the history depth in each time step as $H(s) = s$. One basic realization of the algorithm with $H_{\max}$ reads:

$$H(s) = \min\big\{s,\, H_{\max}\big\}. \tag{7}$$

Due to diffusion it is reasonable to assume that the larger the history depth $H_{\max}$ the better our BC approximates the exact BC.

Based on the general description given in Section 3, the exterior populations of Fig. 4 at the earliest involved time level $s - H(s)$, i.e., $f_i\big(m, s - H(s)\big)$, $m \in \{N + 1, \ldots, N + H(s)\}$ (nodes marked with pentagons in) are assumed to be given. In fact these populations are unknown, and thus finding appropriate populations is

indeed the crucial task of this boundary condition. We preliminary suggest to set those populations according to an equilibrium distribution E_i^{BC}. Several possible ideas for this equilibrium immediately arise, e.g.:

- Initial exterior populations of the original problem:

$$E_i^{BC} = E_i^{\text{ext}} = E_i(\rho_{\text{ext}}, u_{\text{ext}}) \tag{8}$$

- Constant extrapolation of macroscopic boundary quantities:

$$E_i^{BC} = E_i\Big(\rho\big(N, s - H(s)\big), u\big(N, s - H(s)\big)\Big). \tag{9}$$

Of course, other realizations are also conceivable. One further approach is described below.

5. Algorithmic aspects and further refinement

We discuss our DABC as a series of subproblems and its efficient implementation. Furthermore, we aim at enhancing our basic algorithm to improve the efficiency.

5.1. Interpretation as subproblems

For an algorithmic perspective, let all populations (of the computational domain) up to time level $s-1$ be computed. The computation of the inward population $f_1(N, s)$ with history depth $H(s)$ is equivalent to performing $H(s)$ iterations of an LBM subproblem. This subproblem is referred to as the s-th subproblem, see also the dashed box in Fig. 4. We denote the populations of the s-th subproblem by $h_i^s(m, k)$ with $m \in \{0, \ldots, H(s)\}$ and $k \in \{0, \ldots, H(s)\}$. Hence, in the s-th subproblem we have a lattice consisting of $H(s)+1$ nodes, where the time-level starts at $k = 0$. Like in the original problem (f_i's) the populations h_i^s are seen as pre-collision quantities. The subproblem is initialized as

$$h_i^s(0, 0) = f_i\big(N, s - H(s)\big), \tag{10}$$

$$h_i^s(m, 0) = E_i^{BC} \quad \text{for} \quad m \in \{1, \ldots, H(s)\}. \tag{11}$$

Moreover, in this case the interior node at $x = x_N$ ($m = 0$, from the subproblem's view) represents the left boundary of the subproblem. There, the already computed populations are used in a boundary condition which reads:

$$h_i^s(0, k) = f_i(N, s - H(s) + k) \quad \text{for} \quad k \in \{1, \ldots, H(s) - 1\}. \tag{12}$$

A usual collision step is done also in this node after the assignment. In the end, after the last streaming step of the subproblem we obtain the boundary condition for the original problem by

$$f_1(N,s) = h_1^s\big(0, H(s)\big). \tag{13}$$

Note, that the populations at the other boundary of the subproblem for $k \geq 1$ are irrelevant, since they do not influence the final relevant population $h_1^s\big(0, H(s)\big)$. We can even omit the computation of populations at the nodes in the lower triangle within the dashed box in Fig. 4.

For an efficient implementation of the DABC we propose to begin the computation of the s-th subproblem at time-level $s - H(s)$. Then each iteration of the subproblem (LBM with h_i^s) can be done in alignment to an iteration of the original problem (LBM with f_i). By this, one has to handle several subproblems simultaneously, but one inherits the computational benefits of the LBM. We give a more detailed explanation below.

5.2. Error sources of the discrete artificial boundary condition

To measure the accuracy of our DABC, we need a reference solution. To this end, we perform a lattice Boltzmann simulation on a sufficiently extended domain, such that at any investigated time level s the interior and boundary populations are unaffected from the boundary conditions of the extended domain. The corresponding populations shall be denoted by f_i^{ref}. Their initialization has to be done in the same way as f_i and the fictitious exterior nodes (which are real nodes of f_i^{ref}), i.e., by E_i^{ext}. Boundary populations of an ideal (transparent) boundary condition are then given by $f_1^{\mathrm{ref}}(N,s)$, $s \in \mathbb{N}^+$. Thus, we measure the error of the DABC by δ_i:

$$\delta_i(n,s) := \big|f_i(n,s) - f_i^{\mathrm{ref}}(n,s)\big|, \quad (0 \leq n \leq N, \quad s > 0).$$

Notice, if all subproblems were initialized by

$$h_i^s(m,0) = f_i^{\mathrm{ref}}\big(N+m, s-H(s)\big) \quad \text{for} \quad m \in \{1,\ldots,H(s)\},$$

the DABC would be still exact for any possible history depths $H(s)$. Thus, errors are only introduced by assigning non-exact data to the subproblems:

i) the assignment of non-exact initial values in the s-th subproblem, i.e., errors in $h_i^s(m,0)$, $m \in \{1,\ldots,H(s)\}$, and

ii) non-exact initial values in the preceding time levels ($k < s$) led to errors in the boundary populations $f_1(N,k)$ (for $k < s$), which enter the s-th subproblem as boundary conditions, cf. (12).

5.3. Initialization of subproblems' populations

Next we aim at improving the above introduced basic strategy $H(s) = H_{\max}$ for $s \geq H_{\max}$, see (7). To this end, we fix certain initial populations for J successive subproblems. After the corresponding J time steps we will perform a refresh of the initial populations. As we will see, these initial values are given by intermediate results of a certain subproblem with a larger history depth.

Firstly, we consider the s-th subproblem when using a larger history depth $H_2(s)$ than $H(s)$ (i.e., $H_2(s) > H(s)$): The subproblem with history depth $H_2(s)$ can be interpreted as a subproblem with history depth $H(s)$ plus a number of $H_2(s) - H(s)$ additional preceding iterations. Generated by these first $H_2(s) - H(s)$ iterations, the populations at the leftmost nodes, $m = 0, \ldots, H(s)$, are only relevant for the desired value (13). Also, they are sufficient to serve as initial populations in the subproblem with history depth $H(s)$. Thus, taking them as initial populations would finally give the same boundary population $f_1(N, s)$. Meaning, the subproblems with history depths $H(s)$ and $H_2(s)$ are equivalent if the initialization of the subproblem with smaller history depth is done with appropriate intermediate results of the other one.

In this way computed initial populations can also be used for the initialization up to the $(s + J - 1)$-th subproblem provided $H(s + k) \leq H(s)$, $k \leq J - 1$. Notice that information of the leftmost node ($m = 0$) is not used. In fact, it is overwritten by (10) and only (11) is replaced. Thus, J is a refreshing parameter of initial populations, stating how often new initial populations for subproblems are computed. The value $J = 1$ is equivalent to use a larger history depth $H_2(s)$ at all time levels. The initialization of the subproblems with enlarged history depth can for instance be done with the above mentioned possibilities (8)-(9).

In Fig. 5 we illustrate the procedure, where for the first 9 iterations the exact boundary condition $H(s) = s$ is taken. Then, starting at time level $s = 10$ we use a history depth of $H(s) = 5$ and an enlarged history depth $H_2(s) = 10$ together with $J = 7$. This means for every seventh time level two subproblems are considered, one with an enlarged history depth $H_2(s) = 10$ and afterwards one with history depth $H(s)$. The former is simulated for $H_2(s) - H(s)$ iterations and after the final iteration populations are stored. They are used for initialization of subsequent subproblems. For example, the initialization of subproblems 10 (with $H(s)$) to 16 is given by the results after $H_2(10) - H(10) = 5$ iterations of the 10-th subproblem using the enlarged history depth $H_2(10) = 10$.

5.4. Efficient implementation

We have mentioned already above that an efficient implementation will compute several subproblems simultaneously, all aligned by collision and streaming with the

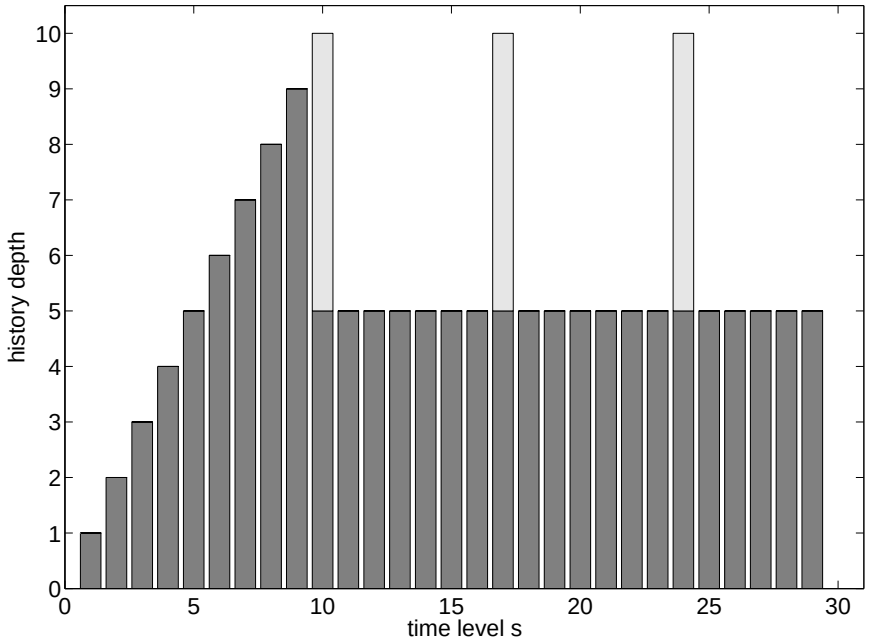


Figure 5: Illustration of history depth $H(s)$: First we have the start-up phase ($s = 1, \ldots, 10$), then initial conditions for the subproblems $s = 11, \ldots, 16$ are fixed. The initial conditions are refreshed at $s = 17$, $s = 24$, and so on. This gives the possibility to better adapt the initial conditions of the subproblems to the actual flow.

original problem. Next, we discuss this in more detail.

Recall that the s-th subproblem yields the inward boundary population at time level s. This is visualized in Fig. 6, where a black box indicates that a certain subproblem (ordinate) gives the inward boundary population of the original problem at the corresponding time level (abscissa). Furthermore, the figure shows that boundary populations of previous time levels enter a certain subproblem depending on the history depth. A box with label j states that information from time level $s-j$ is used in the computation of the s-th subproblem. Hence, the numbers count the (relative) ancestors. The same history depths as in Fig. 5 are used for this illustration. The boxes filled by gray refer to the simulations on an enlarged lattice to compute initial populations. Inspecting a fixed time level in Fig. 6, e.g., $k = 17$, we see that there are multiple subproblems using data from that time level. All these subproblems should be computed simultaneously.

As commonly known, the lattice Boltzmann equation is split into two steps, collision (C) and streaming (S), see also Section 2. And at the boundary points there is a lack of populations after each streaming step. Each boundary condition

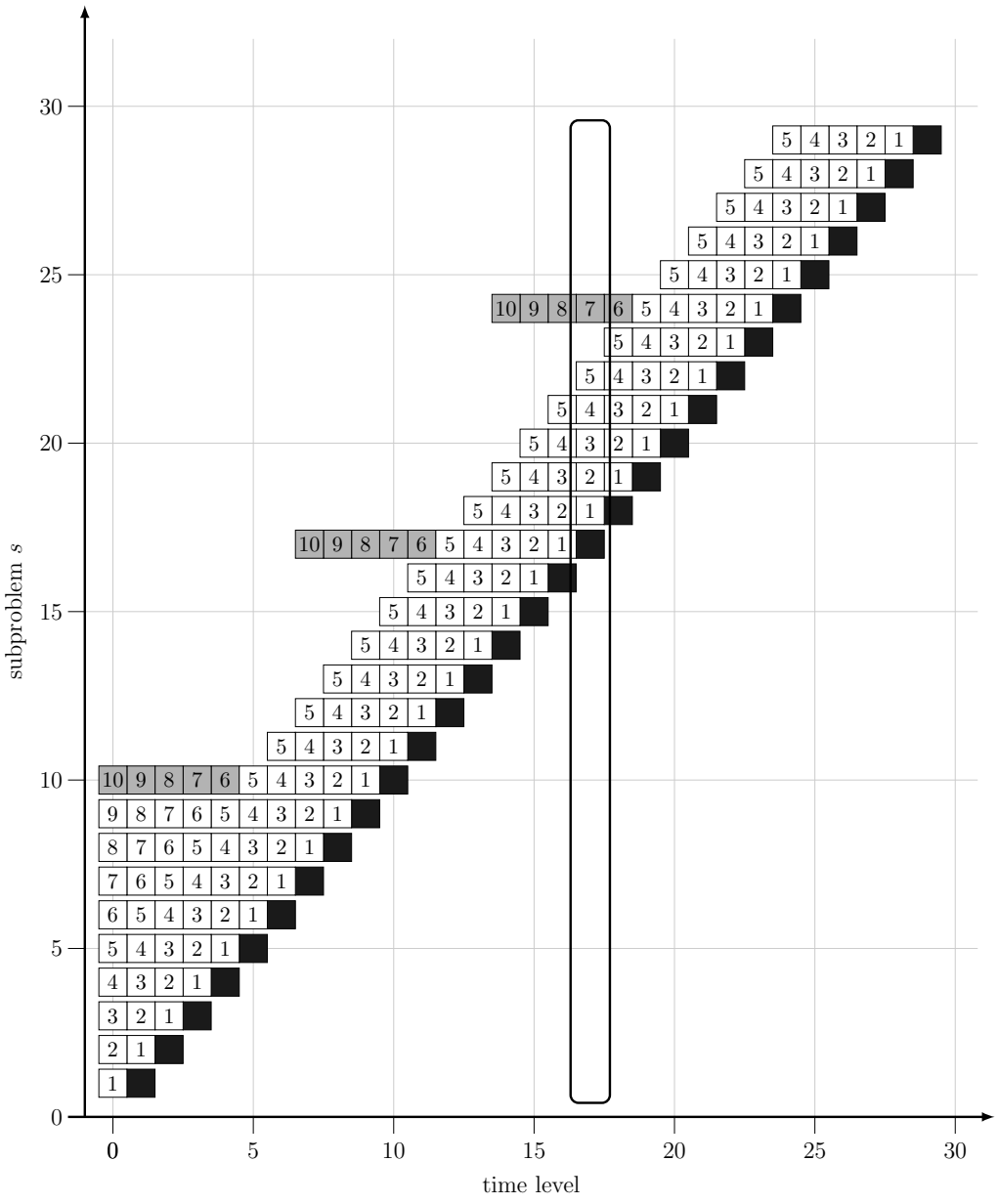


Figure 6: Figure represents different subproblems and illustrates how data of different time levels is used. This shows the potential to simultaneously compute one time step of several subproblems.

(BC) has the task to compensate this lack. Generally, a lattice Boltzmann simulation is structured as

$$\ldots \to C \to S \to BC \to C \to S \to BC \to C \to S \to BC \to \ldots .$$

Now, we focus on the boundary condition steps of this alternating procedure. We have to describe the approach for several subproblems as well as the original problem. For the starting point of the description we assume that the inward population of the original problem is already given. This is true at the beginning of the simulation (by initialization) or at an arbitrary time level after solving the corresponding subproblem. There are several subproblems using data of the current time level, but also additional subproblems have to be created, which depend on the current time level. In the example of Fig. 6 at $k = 17$ the subproblems $18 - 21$ and $24^\star$ exist already, whereas 22 has to be created. The star denotes the subproblem solved to create initial data for later subproblems. For all new subproblems a lattice of corresponding size has to be created and initialized according to (10)-(11). The other subproblems simply copy information from the original problem according to (12). This finishes the boundary condition step. After the subsequent collision and streaming steps, the inward population of the original problem can be copied from the corresponding subproblem, according to (13). Thus, the assumption of the description is again restored.

Given a fixed history depth H, there are H simultaneous subproblems, each having a lattice size of $H + 1$. When neglecting the effort done by copying data, costs of enlarged subproblems and the creation of subproblems, the total effort of the discrete artificial boundary condition is at most as high as a lattice extension by $H^2 + H$ nodes. Thus, the complexity of the DABC is determined by the history depth H.

6. Numerical results

For qualitative statements we compare results of the DABC with those by non-reflecting *characteristic boundary conditions* (CBCs) [13, 14] as well as those by an *impedance boundary condition* (IBC) [15].

The non-reflecting CBC for a right boundary in 1D is summarized as follows: In each iteration, we first compute at the right boundary node the wave amplitude variation

$$\mathcal{L} = (u + c_s) \left[c_s^2 \frac{\partial \rho}{\partial x} + c_s \rho \frac{\partial u}{\partial x} \right], \tag{14}$$

where $c_s = \frac{1}{\sqrt{3}}$ denotes the speed of sound in D1Q3. Then a Dirichlet boundary condition for macroscopic quantities is imposed by solving the ODE system

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ u\end{pmatrix} = -\mathcal{L}\begin{pmatrix}(2c_s^2)^{-1}\\ (2\rho c_s)^{-1}\end{pmatrix}. \tag{15}$$

In the simulations below we use a one-sided second order finite difference quotient for the derivatives in the wave amplitude variation (14). Moreover, we solve (15) by using an explicit Euler scheme. Here, the corresponding boundary populations $f_i^{\mathrm{CBC}}(N,\cdot)$ are obtained from an evaluation of the equilibrium distribution (3) employing the numerical approximation of (15) for ρ and u. This transfer can be improved, e.g., by using numerical lifting operators [23].

The impedance boundary condition (IBC) of Schlaffer [15] is derived under the assumption that any pressure variation travels with a velocity of $|c_s|$. Unlike the CBC, only the unknown population is computed. The inward population at a right boundary for time level s is determined by

$$f_1^{\mathrm{IBC}}(N,s) = \frac{(1-\bar{u})f_2^{\mathrm{IBC}}(N,s) - f_0^{\mathrm{IBC}}(N,s)\bar{u}}{1+\bar{u}}.$$

Here, the velocity $\bar{u}$ is derived from a balance equation for the incoming momentum flux and the rate of momentum change in a control volume, such that both terms compensate one another. That is

$$\begin{aligned}\bar{u} = u(N,s-1) &+ c_s^2\rho_z + c_s\\ &- \sqrt{(c_s^2\rho_z + c_s)^2 + 2c_s^2\rho_z\big[1+u(N,s-1)\big] - 2c_s^2}\,,\end{aligned}$$

where

$$\rho_z = \frac{\rho(N,s-1)}{f_0^{\mathrm{IBC}}(N,s) + 2f_2^{\mathrm{IBC}}(N,s)}.$$

Although populations enter the above computation, we emphasize that the IBC is not derived from discrete considerations. See [15] for further details.

There is no free parameter controlling the complexity neither in the CBC nor in the IBC. Hence, we see that in contrast to the DABC both, the CBC and the IBC, have a fixed effort.

6.1. Simple density pulse

For the first numerical test, we consider a Cauchy problem with initial data for the mass density (at $t = 0$, cf. [24])

$$\rho_0(x) = \begin{cases} 1 & \text{for} \quad x \leq 0.3, \\ 1 + 0.4 \cdot \exp\left(\frac{-15^{-2}}{(x-0.3)^2}\right) \cdot \exp\left(\frac{-15^{-2}}{(x-0.7)^2}\right) & \text{for} \quad 0.3 < x < 0.7, \\ 1 & \text{for} \quad 0.7 \leq x, \end{cases}$$

such that $\rho_0 \in C^\infty(\mathbb{R})$. The fluid velocity is initialized uniformly by u_0. To evaluate our DABC we introduce a finite spatial domain $\mathcal{I} := [-10, 1]$, such that we have effects of the artificial boundary only at $x = 1$ in the first iterations of the LBM. To this end, the interval $\mathcal{I}$ is discretized with step size $h = 0.005$. Hence, the computational grid is given by $G_x := \{x_0, x_1, \ldots, x_N\}$ with $N = 11/h$. Then corresponding initial populations in the computational domain are computed by an evaluation of the equilibrium distribution (3):

$$f_i(n, 0) = w_i \rho_0(x_n) \left[1 + 3c_i u_0 + \frac{9}{2}(c_i u_0)^2 - \frac{3}{2}u_0^2\right].$$

To begin, we illustrate the evolution of the signal in a simulation with $u_0 = 0$, $\tau = 1$ and history depth (7) with $H_{\max} = 20$. All subproblems are initialized by (8) or (9), respectively. The initialization strategy given in Section 5 is not applied, yet. In Figs. 7 and 8 the density and velocity profiles are plotted, respectively. Note that the scaling of the y-axes are adapted.

One can see that due to the initial profile, two pressure waves are generated, one traveling in negative and the other in positive direction, respectively. We clearly see some reflection generated at the boundary, the reflected wave is traveling leftwards into the computational domain. The ideal level of the signals (density and velocity) after the pulse has left the computational domain are reached neither by the IBC nor the CBC nor the DABC with (8), which can be seen best at the plots corresponding to time level 500. The DABC with (9) is much closer to the ideal level, however it has a larger reflection. What can be hardly seen in these plots is a tiny oscillating behavior of the DABC with (9). This becomes more pronounced when enlarging the history depth. A corresponding plot for $H_{\max} = 60$ is given in Fig. 9. Notice, the enlargement of $H_{\max}$ reduces the error peak of the reflection but increases the oscillations for the DABC with (9).

To further investigate the effect of the history depth in the DABC in more detail, we measure the spatial absolute error:

$$\mathrm{AbsErr}_z(t) := \max_{x \in G_x} |z(x, t) - z_{\mathrm{ref}}(x, t)|, \tag{16}$$

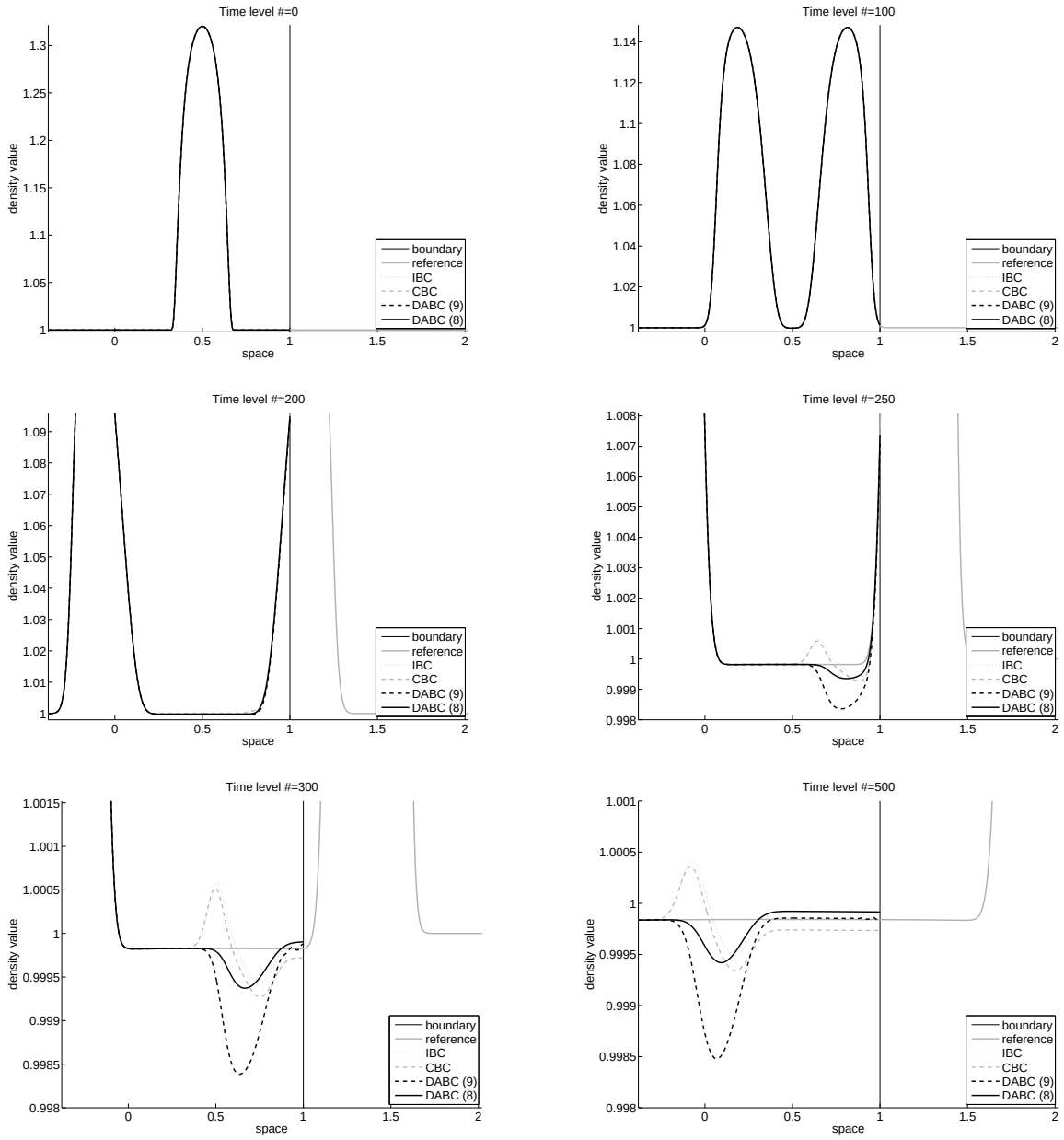


Figure 7: Density profile at time step 0, 100, 200, 250, 300, 500: using history depth (7) with $H_{\max} = 20$ (scaling of the y-axis is adapted): From the time on, when the density wave hits the artificial boundary (time level 100), artificial reflections occur for all approximations (see time level 200 following). CBC and IBC initiate a wave (with peak and trough) during the interaction. In contrast, the DABC introduces merely a trough (on the largest error scale).

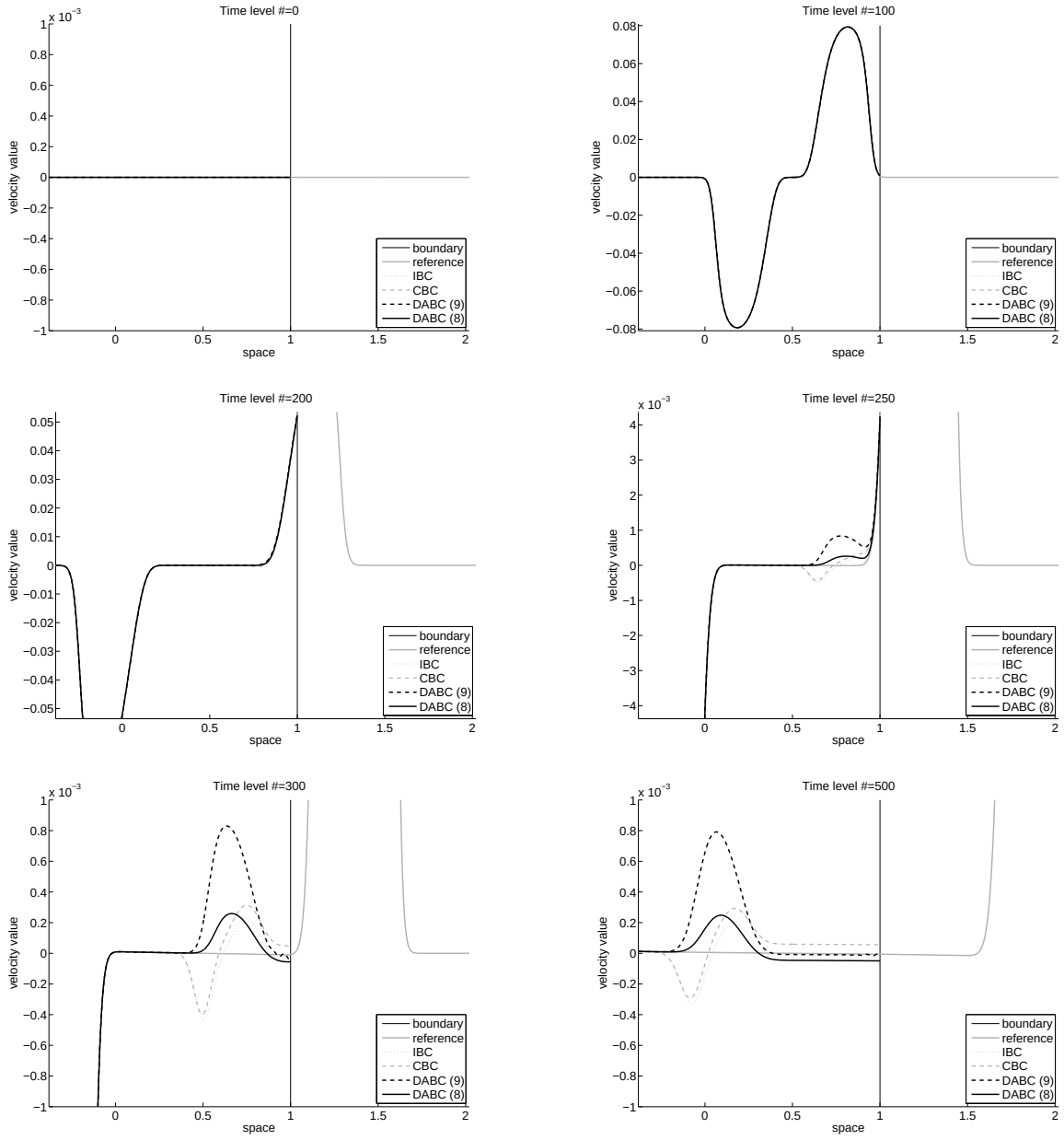


Figure 8: Velocity profile at time step 0, 100, 200, 250, 300, 500: using history depth (7) with $H_{\max} = 20$ (scaling of the y-axis is adapted): One can observe similar reflections as for the density.

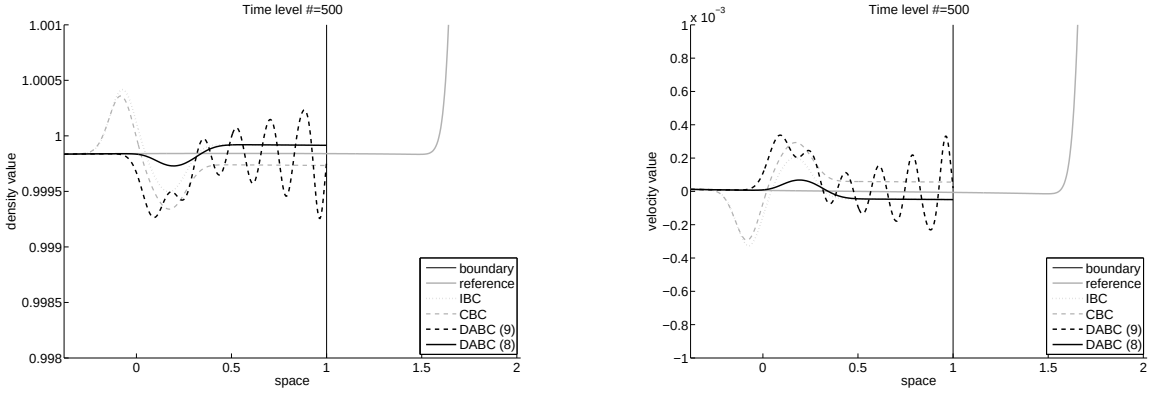


Figure 9: Density profile (left) and velocity profile (right) at time step 500. For the DABC $H_{\max} = 60$ is used.

where z is either ρ or u. The absolute errors (16) of the DABC for different history depths $H(s) = \min\{s, H_{\max}\}$ with $5 \leq H_{\max} \leq 75$ are visualized in Fig. 10. We can see that a larger history depth reduces the errors. Also the errors of the CBC and the IBC are shown (transparent surfaces), demonstrating that the DABC with (8) is superior starting from a certain history depth.

From the errors shown in Fig. 10 it is reasonable to concentrate on the maximal error:

$$\mathrm{Err}_z := \max_t \{\mathrm{AbsErr}_z(t)\}$$

where z is again either ρ or u. By this simplified view we can more easily visualize the dependencies on τ and u_0. The plots in Fig. 11 show this maximal error for different combinations of history depths $H(s)$ and velocities u_0. We see that for all velocities u_0 the error can be decreased by increasing the maximal history depth. Similarly, the plots in Fig. 12 show the influence of the relaxation parameter τ to the maximal error. Here the same positive effect of the maximal history depth to the error can be observed.

Up to now we have not made use of the initialization strategy given in Section 5 in our numerical tests. To demonstrate its influence, we consider the test case with $\tau = 1.4$ and $u_0 = 0$. As reference values we use the DABC with (8) without the initialization strategy of Section 5. The corresponding maximal errors are shown for different history depths in Fig. 13 by the thin gray line. It clearly shows oscillations, the thick gray line is the average of two consecutive values. For the range of history depths we repeat the simulations using the initialization strategy of Section 5. The dashed lines show the averaged maximal errors with an enlarged history depth $H_2 =$

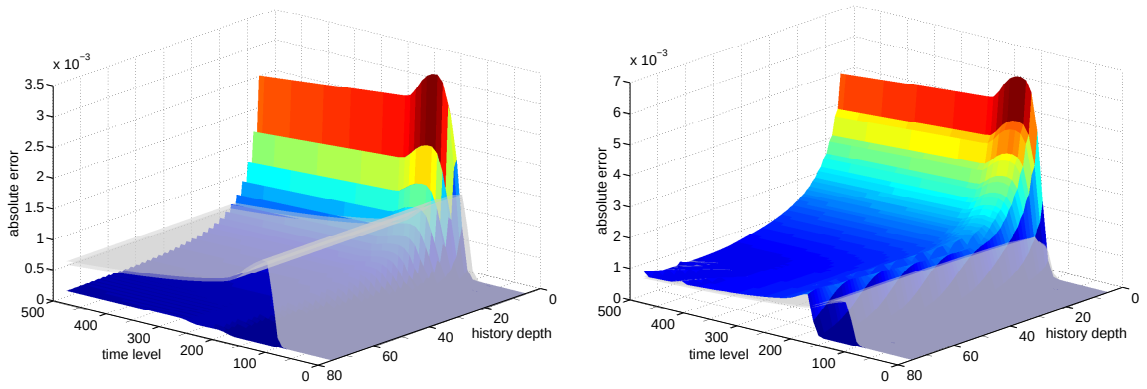


Figure 10: Absolute errors in the density: DABC with (8) (left) and with DABC (9) (right). The transparent surface gives the absolute error of the CBC. Errors in the velocities behave similarly. There is an even-odd (zig-zag) behavior. Thus, one should use odd history depths.

$H+10$, whereas the solid black lines refer to the choice $H_2 = 2H$. The two selections $J = 5$ and $J = 20$ for the refresh parameter are chosen for illustration. Note, that for clarity the lines show only the average of two consecutive values, since oscillations are also present in the other data. The maximal errors are always smaller than the reference values except for odd history depths in the case $H_2 = 2H$ and $J = 20$.

6.2. Acoustic sinusoidal signal

In the second numerical test an acoustic wave is temporary generated by a point source at $x = -1$. We follow the simple approach given in [25]. The populations at the corresponding source node are computed by the equilibrium distribution (3) determined by velocity $u(0,t) = 0$ and an oscillating density according to

$$\rho(0,t) = 1 + \rho_{\text{src}} \sin\left(\frac{2\pi}{T}t\right). \tag{17}$$

The period of the waves is determined by T and the amplitude is controlled by ρ_{src}. It is a drawback of this approach that the real amplitude is not equal to ρ_{src}. A more sophisticated and effective approach for generating acoustic waves is presented by the same author in [26]. For our testing purposes the simple approach is sufficient and we employ the point source (17) with parameters $\rho_{\text{src}} = 0.03$ and $T = 100$.

For the spatial domain, we consider the interval $\mathcal{J} := [-5, 1]$ with step size $h = 0.005$. At the right boundary at $x = 1$ the DABC is applied. The point source is activated from the beginning of the simulation up to time level 1000 and we use in this section always $\tau = 1$. The first variations generated from the point source arrive

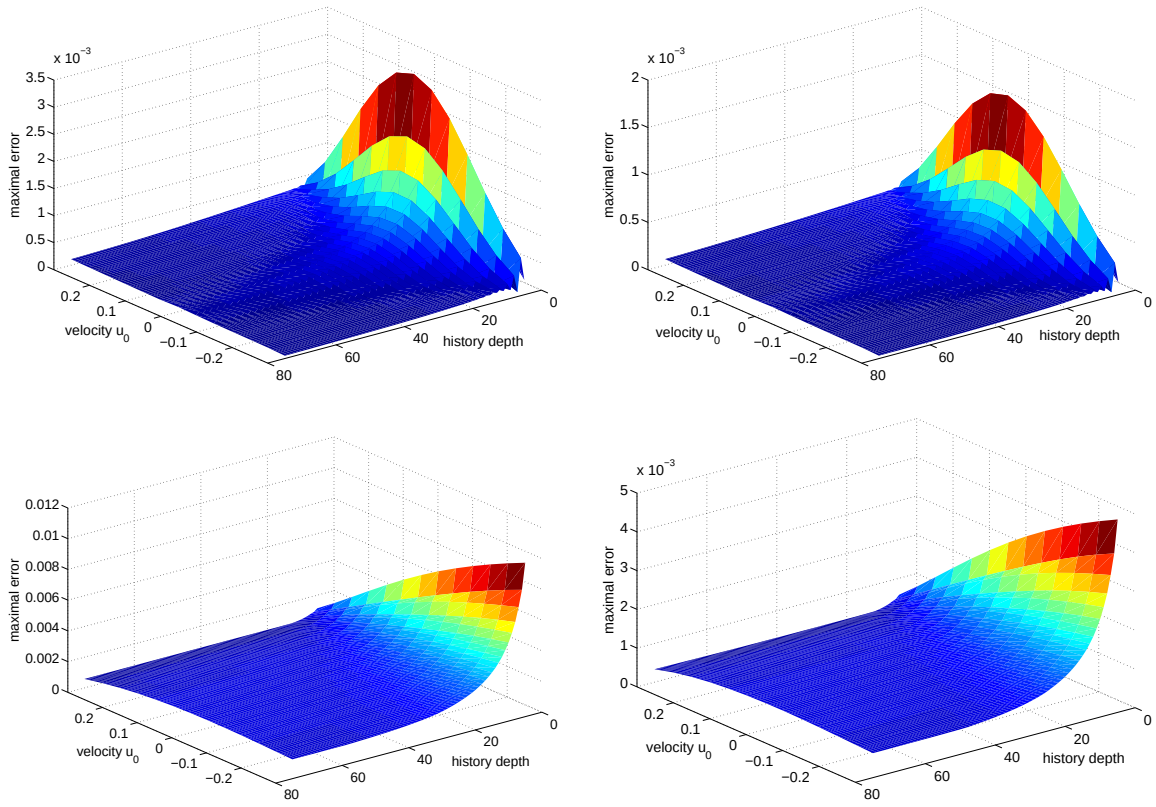


Figure 11: Maximal error in density (left) and velocity (right) for different history depths $H(s)$ and velocities u_0 ($\tau = 1$). Plots in the top row correspond to the DABC with (8) and with (9) in the bottom row. Clearly, the higher u_0, the faster the wave travels through the artificial boundary and the smaller the errors. Notice, the error in DABC with (8) decreases also for decreasing negative velocities u_0.

at the right boundary not before time level 400, see also Fig. 14 for an illustration of the test case. The plots show the reference solution computed on a sufficiently larger domain. Moreover the errors of the IBC, CBC and DABC are plotted. We clearly see that the DABC with (9) is not stable. We like to mention that the instabilities (in this test) are related to the choice of $H_{\max}$, when taking odd numbers, e.g., $H_{\max} = 25$, no instabilities are observed. This needs further investigation, however by (8) we have an initialization with a generally better behavior than by (9). Similarly, a qualitative different behavior for even/odd history depths is also visible for initialization by (8) above, e.g., Fig. 13. We conclude again that the initialization of the subproblems is very crucial and that odd history depths should be preferred.

To underline the effect of the history depth, we consider the absolute error (16) for

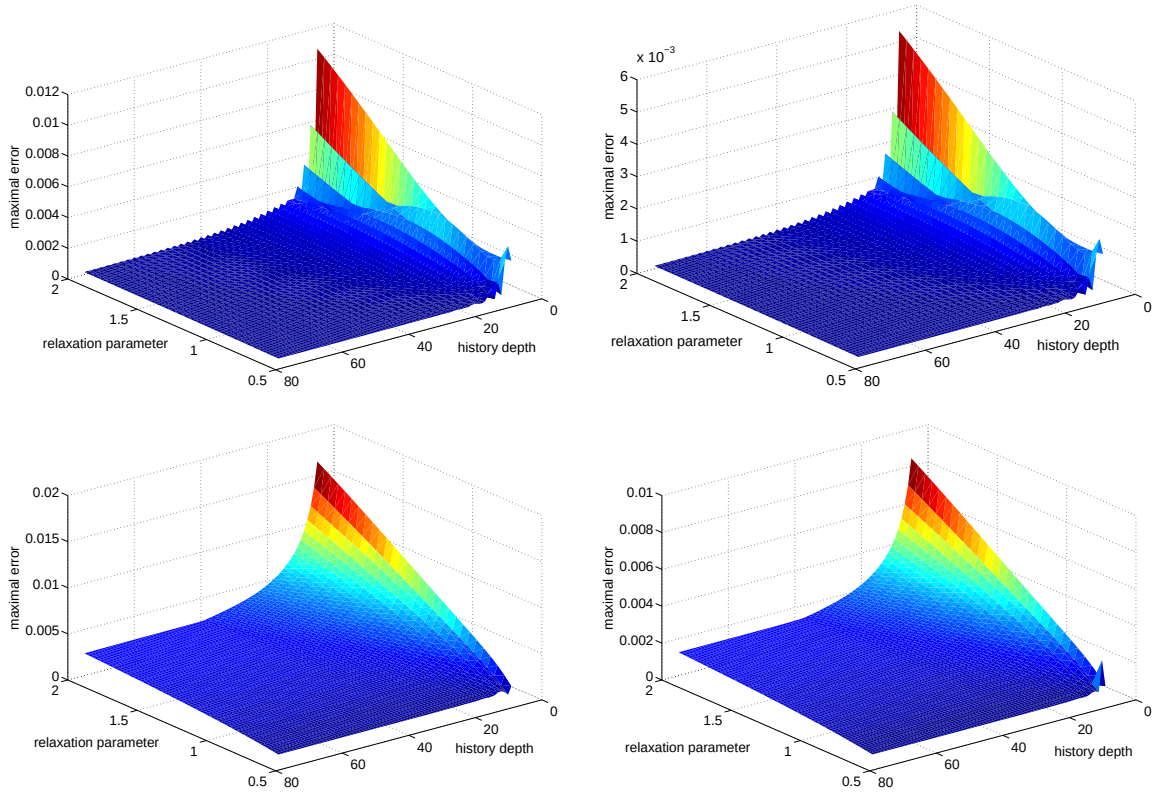


Figure 12: Maximal error in density (left) and velocity (right) for different history depths $H(s)$ and relaxation parameters τ ($u_0 = 0$). Plots in the top row correspond to the DABC with (8) and with (9) in the bottom row. — The smaller the relaxation parameter τ the smaller the error.

different history depths. We omit the DABC with (9), due to its instable behavior. Fig. 15 indicates that the maximal error is reduced when $H_{\max}$ is increased. We also tested different parameter combinations, which all confirm a reduced error by a larger history depth.

7. Conclusions

For D1Q3 in LBM, we explained a tree interpretation for the evolution of populations. This allowed us to formulate a discrete artificial boundary condition (DABC) based on fictitious nodes in the exterior domain and based on a variable history depth. We showed that these DABCs are equivalent to solve LBM subproblems. A key role in the DABC approach is the initialization of corresponding subproblems. For this task we introduced two natural possibilities, whereof only one showed to be

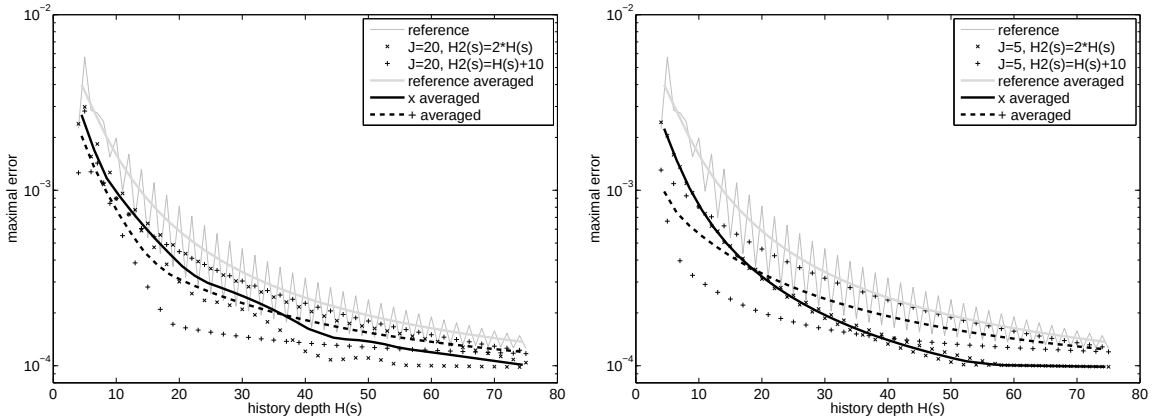


Figure 13: Maximal errors for initialization strategy of Section 5. For the left plot the refresh parameter is $J = 20$, and $J = 5$ for the right one. For the reference line, the maximal error of an odd history depth is significantly smaller than the two adjacent even history depths.

suitable, based on the results of numerical tests. We briefly discussed the possibility of an efficient implementation.

The two numerical tests demonstrated that the history depth controls the accuracy of the DABC. Moreover, the tests showed that we can improve upon the characteristic boundary condition (CBC) as well as the impedance boundary condition (IBC) in terms of accuracy. For achieving the same accuracy, the CBC and IBC have a smaller effort. However the DABC has the advantage of an adjustable accuracy, which is aligned with a changing complexity. In future work, we will extend our DABC to LBM models for higher spatial dimensions.

References

[1] S. Chen, G. D. Doolen, Lattice Boltzmann Method for Fluid Flows, Annual Review of Fluid Mechanics 30 (1) (1998) 329–364.

[2] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford University Press, Oxford, UK, 2001.

[3] D. Heubes, A. Bartel, M. Ehrhardt, An Introduction to the Lattice Boltzmann Method for Coupled Problems, pp. 3–30 in [27].

[4] D. Haydock, J. M. Yeomans, Lattice Boltzmann simulations of acoustic streaming, Journal of Physics A: Mathematical and General 34 (25) (2001) 5201–5213.

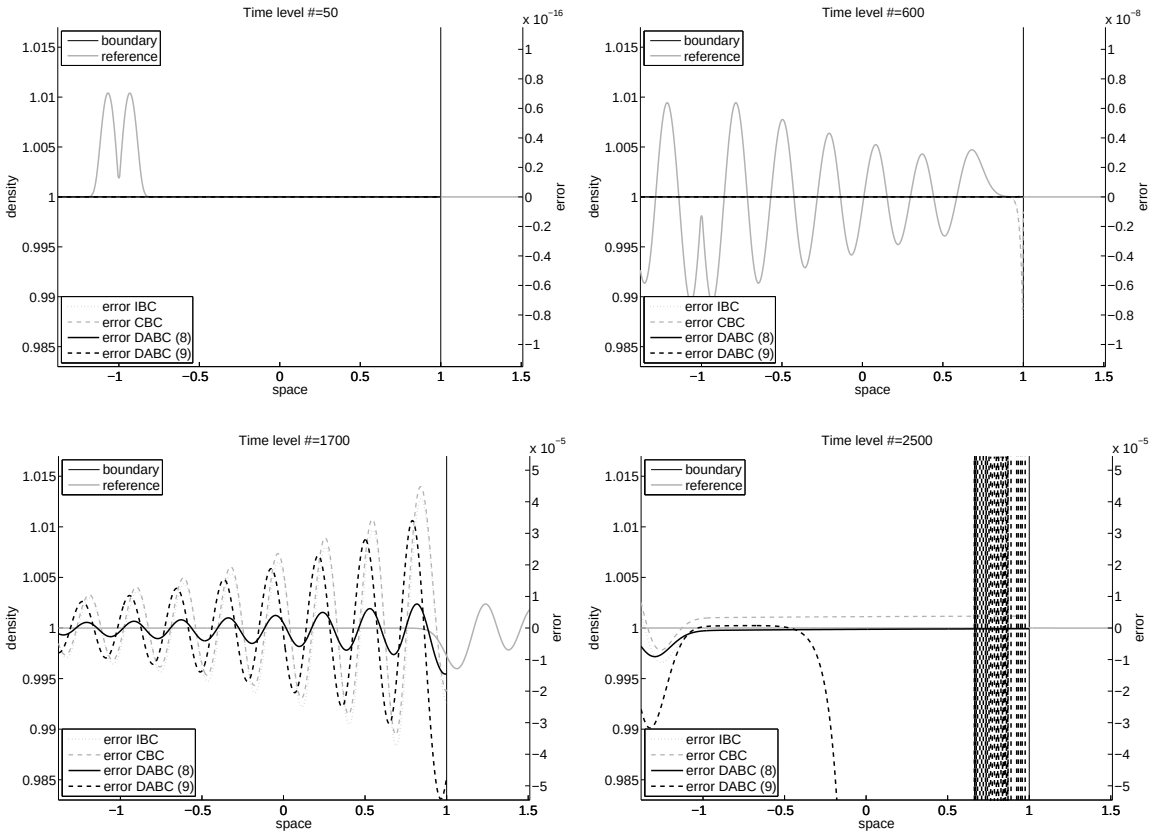


Figure 14: Evolution of the reference density profile ρ_{ref} in the acoustic test example (left ordinate axis); errors of different boundary conditions (right ordinate axis); $H_{\max} = 18$.

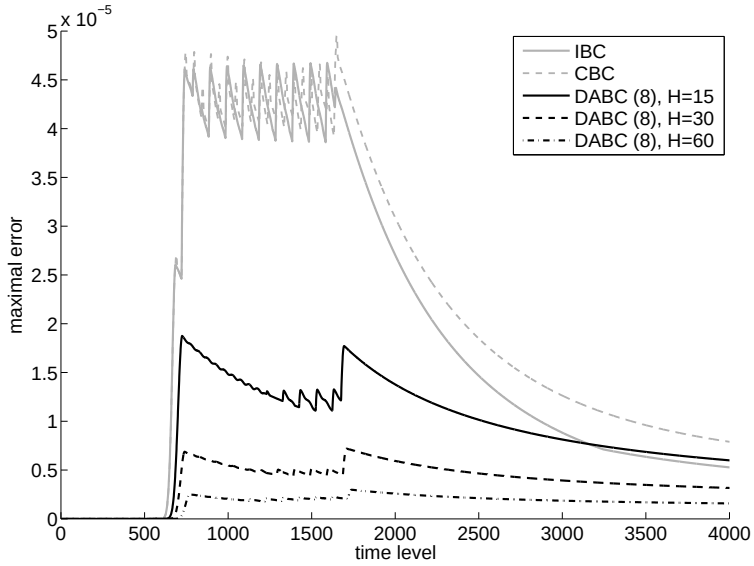


Figure 15: Evolution of the absolute error for the acoustic test example.

[5] C. Sun, L. L. Munn, Lattice-Boltzmann simulation of blood flow in digitized vessel networks, Computers & Mathematics with Applications 55 (7) (2008) 1594–1600.

[6] S. Geller, J. Tölke, M. Krafczyk, Lattice-Boltzmann Method on Quadtree-Type Grids for Fluid-Structure Interaction, pp. 270–293 in [28].

[7] B. Engquist, A. Majda, Absorbing boundary conditions for the numerical simulation of waves, Mathematics of Computation 31 (1977) 629–651.

[8] G. W. Hedstrom, Nonreflecting boundary conditions for nonlinear hyperbolic systems, Journal of Computational Physics 30 (2) (1979) 222–237.

[9] K. W. Thompson, Time dependent boundary conditions for hyperbolic systems, Journal of Computational Physics 68 (1) (1987) 1–24.

[10] D. Kröner, Absorbing boundary conditions for the linearized Euler equations in 2-D, Mathematics of Computation 57 (195) (1991) 153–167.

[11] T. Poinsot, S. Lele, Boundary conditions for direct simulations of compressible viscous flows, Journal of Computational Physics 101 (1) (1992) 104–129.

[12] M. Ehrhardt, Absorbing boundary conditions for hyperbolic systems, Numerical Mathematics: Theory, Methods and Applications 3 (3) (2010) 295–337.

[13] S. Izquierdo, N. Fueyo, Characteristic nonreflecting boundary conditions for open boundaries in lattice Boltzmann methods, Physical Review E 78 (2008) 046707.

[14] D. Kim, H. M. Kim, M. S. Jhon, S. J. Vinay III, J. Buchanan, A characteristic non-reflecting boundary treatment in lattice Boltzmann method, Chinese Physics Letters 25 (6) (2008) 1964.

[15] M. Schlaffer, Non-reflecting boundary conditions for the lattice Boltzmann method, PhD thesis, TU München, Germany, 2014.

[16] A. Najafi-Yazdi, L. Mongeau, An absorbing boundary condition for the lattice Boltzmann method based on the perfectly matched layer, Computers & Fluids 68 (2012) 203–218.

[17] M. Tekitek, M. Bouzidi, F. Dubois, P. Lallemand, Towards perfectly matching layers for lattice Boltzmann equation, Computers & Mathematics with Applications 58 (5) (2009) 903–913.

[18] J.-P. Bérenger, A perfectly matched layer for the absorption of electromagnetic waves, Journal of Computational Physics 114 (2) (1994) 185–200.

[19] D. Heubes, A. Bartel, M. Ehrhardt, Exact artificial boundary conditions for a lattice Boltzmann method, Computers & Mathematics with Applications 67 (11) (2014) 2041–2054.

[20] D. Heubes, A. Bartel, M. Ehrhardt, Discrete artificial boundary conditions for the lattice Boltzmann method in 2D, Preprint BUW-IMACM 14/37, Universität Wuppertal, Germany, 2014.

[21] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Physical Review 94 (3) (1954) 511–525.

[22] Y. Qian, D. d'Humieres, P. Lallemand, Lattice BGK Models for Navier-Stokes Equation, Europhysics Letters 17 (6) (1992) 479–484.

[23] Y. Vanderhoydonc, Numerical lifting operators for kinetic Boltzmann models, PhD thesis, Universiteit Antwerpen, Belgium, 2014.

[24] M. Junk, M. Rheinländer, Regular and multiscale expansions of a lattice Boltzmann method, Progress in Computational Fluid Dynamics 8 (2006) 25–37.

[25] E. M. Viggen, The lattice Boltzmann method in acoustics, in: 33rd Scandinavian Symposium on Physical Acoustics, 2010.

[26] E. M. Viggen, Acoustic multipole sources for the lattice Boltzmann method, Physical Review E 87 (2) (2013) 023306.

[27] M. Ehrhardt (Ed.), Progress in Computational Physics: Novel Trends in Lattice-Boltzmann Methods - Reactive Flow, Physicochemical Transport and Fluid-Structure Interaction, Volume 3, Bentham Science Publishers Ltd., 2013.

[28] H.-J. Bungartz, M. Schäfer, Fluid-Structure Interaction: Modelling, Simulation, Optimisation, Volume 53 of Lecture Notes in Computational Science and Engineering, Springer Verlag, 2006.